























VelocityDb Namespace



The **VelocityDb** name space contains classes for the VelocityDB Object Database.

Classes

| Class | Description |
|------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllObjects(T) | An object of this type is returned by AllObjects(T)(Boolean, Boolean) or AllObjects(T)(Boolean) |
|  AllObjects(T).AllObjectsIterator | Iterator of type AllObjects(T) |
|  AllObjectsExtension | A few extensions to improve performance of Linq for Objects queries |
|  AutoPlacement | This is the placement manager of objects persisted using Persist(Object) , it is automatically created when the first object of a type is persisted. It is the first object in the database of each type mapped Database. |
|  Database | Represents a Database which corresponds to file system file. |
|  DatabaseLocation | Represents a host and a directory in which a range of databases are stored. The range is a database number range. |
|  DatabaseLocations | Maintains collections of all existing DatabaseLocations . |
|  DataCache | Object maintains a list of pages accessed by a session, pages are removed from the list when there is not enough memory available. This makes such pages eligible for garbage collection while in list pages are prevented from being garbage collected. |
|  License | Contains a list of possible license attributes for licensing VelocityDb class library and server |
|  OfType | An object of this type is a base class of type returned by AllObjects(T)(Boolean, Boolean) or AllObjects(T)(Boolean) |
|  OfType.OfTypeIterator | Iterator for type OfType |
|  OfTypeExtension | A few extensions to improve performance of Linq for Objects queries |
|  OptimizedPersistable | Base class for all persistent capable classes except for embedded objects which do not need to be subclasses of this class. |
|  Page | Each Database consist of a number of variable sized pages. A page can be compressed and may be encrypted and contains one or more objects. Page compression is controlled by the DatabaseLocation of |

| | | |
|--------------------------------------------------------------------------------------------------------------------------------------------|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | the Database containing a page. Each page has a PageInfo that provides info about a page and can be updated to turn on/off compression and encryption for a specific page. |
|  PageCache | | Maintains string references to Pages within a Database |
|  PageInfo | | Each Page has a PageInfo that provides info about a page and can be used to turn on/off compression and encryption for a page. |
|  PageOffset | | Internally Used within VelocityDB and its extensions |
|  Placement | | This class is used when deciding where to place a new object persistently. That is we have a choice of database number, page number and page number. There is other ways to control the placement including ObjectsPerPage , PagesPerDatabase , and it is also possible to override Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) . |
|  ReferenceTracked | | Tracks references to this object and signals ReferentialIntegrityException if unpersisted before while still being referenced. |
|  WeakIOptimizedPersistableReference(T) | | When a persistent object is opened, all its referenced objects are opened as well. In order to limit the number of opened objects, you need to use this class. A reference from an instance of this class will not cause the referenced object to be opened. It will be opened when you ask for it. |
|  WeakIOptimizedPersistableReferenceBase | | This is the base class for all weak references. |
|  WeakIOptimizedPersistableReferenceX(T) | | Obsolete. Deprecated, use WeakIOptimizedPersistableReference instead (so name conflict can be avoided) When a persistent object is opened, all its referenced objects are opened as well. In order to limit the number of opened objects, you need to use this class. A reference from an instance of this class will not cause the referenced object to be opened. It will be opened when you ask for it. The Type of the weak referenced object, must be OptimizedPersistable |

Structures

| Structure | Description |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Oid | The object identifier containing a 32 bit database part, a 16 bit page part and a 16 bit page part. These parts are combined into a 64bit unsigned number Id . |
|  OidShort | The short object identifier containing a 16 bit page part and a 16 bit page part. This is used for short references within a single databases. |

Interfaces

| | Interface | Description |
|---|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| → | IOptimizedPersistable | add optimized persistence by implementing this interface. We provide OptimizedPersistable as a base class that implements this interface. |
| → | IReferenceTracked | Maintains a set of all References to this object |

Enumerations

| | Enumeration | Description |
|---|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| → | CacheEnum | Choices for object caching |
| → | OpFlags | Flag bits for a persistent object |
| → | PageInfo.compressionKind | Selection of choices for Page compression. |
| → | PageInfo.encryptionKind | Use desEncrypted for encrypted pages, set encryption private key in DatabaseLocation . Use noEncryption when no encryption is desired. Other choices in this enum are not yet publicly available. Other kinds of encryption can be provided upon request. We can even make the interface public allowing application defined encryption. |

AllObjects(T) Class

An object of this type is returned by [AllObjects\(T\)\(Boolean, Boolean\)](#) or [AllObjects\(T\)\(Boolean\)](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OfType](#)

VelocityDb.AllObjects(T)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class AllObjects<T> : OfType,  
    IEnumerable<T>, IEnumerable
```

VB

```
Public Class AllObjects(Of T)  
    Inherits OfType  
    Implements IEnumerable(Of T), IEnumerable
```

C++

```
generic<typename T>  
public ref class AllObjects : public OfType,  
    IEnumerable<T>, IEnumerable
```

F#

```
type AllObjects<'T> =  
    class  
        inherit OfType  
        interface IEnumerable<'T>  
        interface IEnumerable  
    end
```


Type Parameters



T

The object type we are looking for in the iteration (enumeration)


The AllObjects(T) type exposes the following members.

Constructors


| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------|
|  | AllObjects(T)(BTreeSet(T)) | Constructs this enumeration wrapper object |

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------|
|  | AllObjects(T)(Database, Boolean) | Constructs this enumeration wrapper object |
|  | AllObjects(T)(SessionBase, Boolean, Boolean) | Constructs this enumeration wrapper object |






Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|--------------------------------------------|
|  | Count | Get a count of instances of the given type |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------|
|  | GetEnumerator | Enumeration of all objects of type T |

Extension Methods




| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Count(T) | Override to improve performance over IEnumerable LINQ extension (Defined by AllObjectsExtension .) |
|  | ElementAt(Int32) | Overloaded. Override LINQ for faster access (Defined by OfTypeExtension .) |
|  | ElementAt(T)(Int32) | Overloaded. Override LINQ for faster access (Defined by AllObjectsExtension .) |
|  | Skip(Int32) | Overloaded. Bypasses a specified number of elements in a sequence and then returns the remaining elements. (Defined by OfTypeExtension .) |
|  | Skip(TSource)(Int32) | Overloaded. Bypasses a specified number of elements in a sequence and then returns the remaining elements. Overrides Skip(TSource)(IEnumerable(TSource), Int32) for improved performance (Defined by AllObjectsExtension .) |

See Also

[VelocityDb Namespace](#)

AllObjects(T) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------|
|  | AllObjects(T)(BTreeSet(T)) | Constructs this enumeration wrapper object |
|  | AllObjects(T)(Database, Boolean) | Constructs this enumeration wrapper object |
|  | AllObjects(T)(SessionBase, Boolean, Boolean) | Constructs this enumeration wrapper object |

See Also

[AllObjects\(T\)Class](#)

[VelocityDb Namespace](#)

AllObjects(T) Constructor (BTreeSet(T))

Constructs this enumeration wrapper object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AllObjects (  
    BTreeSet<T> bTree  
)
```

VB

```
Public Sub New (  
    bTree As BTreeSet(Of T)  
)
```

C++

```
public:  
AllObjects (  
    BTreeSet<T>^ bTree  
)
```

F#

```
new :  
    bTree : BTreeSet<'T> -> AllObjects
```

Parameters

bTree

Type: [VelocityDb.Collection.BTree.BTreeSet\(T\)](#)

A collection

See Also

[AllObjects\(T\)Class](#)

[AllObjects\(T\)Overload](#)

[VelocityDb Namespace](#)

AllObjects(T) Constructor (Database, Boolean)

Constructs this enumeration wrapper object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AllObjects(  
    Database db,  
    bool includeSubclasses = false  
)
```

VB

```
Public Sub New (  
    db As Database,  
    Optional includeSubclasses As Boolean = false  
)
```

C++

```
public:  
AllObjects(  
    Database^ db,  
    bool includeSubclasses = false  
)
```

F#

```
new :  
    db : Database *  
    ?includeSubclasses : bool  
(* Defaults:  
    let_includeSubclasses = defaultArg includeSubclasses false  
)  
-> AllObjects
```

Parameters

db

Type: [VelocityDb.Database](#)

The active db

includeSubclasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes

See Also

[AllObjects\(T\)Class](#)

[AllObjects\(T\)Overload](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

AllObjects(T) Constructor (SessionBase, Boolean, Boolean)

Constructs this enumeration wrapper object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AllObjects(  
    SessionBase session,  
    bool includeSubclasses,  
    bool databasePerType  
)
```

VB

```
Public Sub New (  
    session As SessionBase,  
    includeSubclasses As Boolean,  
    databasePerType As Boolean  
)
```

C++

```
public:  
AllObjects(  
    SessionBase^ session,  
    bool includeSubclasses,  
    bool databasePerType  
)
```

F#

```
new :  
    session : SessionBase *  
    includeSubclasses : bool *  
    databasePerType : bool -> AllObjects
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

includeSubclasses

Type: [System.Boolean](#)

Also return instances of sub classes

databasePerType

Type: [System.Boolean](#)

VelocityDB Class Library

Assume that persisted objects were made persistent the simple way using [Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#). When persisting this way, each object type gets its own [Database](#) which makes finding these objects easier and faster

See Also

[AllObjects\(T\)Class](#)


[AllObjects\(T\)Overload](#)

[VelocityDb Namespace](#)

AllObjects(T).AllObjects(T) Properties

The [AllObjects\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|--------------------------------------------|
|  | Count | Get a count of instances of the given type |

See Also

[AllObjects\(T\)Class](#)

[VelocityDb Namespace](#)

AllObjects(T).Count Property

Get a count of instances of the given type

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Count { get; }
```

VB

```
Public ReadOnly Property Count As ULong  
    Get
```

C++

```
public:  
property unsigned long long Count {  
    unsigned long long get ();  
}
```

F#

```
member Count : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also


[AllObjects\(T\)Class](#)

[VelocityDb Namespace](#)






AllObjects(T).AllObjects(T) Methods

The [AllObjects\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------|
|  | GetEnumerator | Enumeration of all objects of type T |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Count(T) | Override to improve performance over IEnumerable LINQ extension (Defined by AllObjectsExtension.) |
|  | ElementAt(Int32) | Overloaded. Override LINQ for faster access (Defined by OfTypeExtension.) |
|  | ElementAt(T)(Int32) | Overloaded. Override LINQ for faster access (Defined by AllObjectsExtension.) |
|  | Skip(Int32) | Overloaded. Bypasses a specified number of elements in a sequence and then returns the remaining elements. (Defined by OfTypeExtension.) |
|  | Skip(TSource)(Int32) | Overloaded. Bypasses a specified number of elements in a sequence and then returns the remaining elements. Overrides Skip(TSource)(IEnumerable(TSource), Int32) for improved performance (Defined by AllObjectsExtension.) |

See Also

[AllObjects\(T\)Class](#)

[VelocityDb Namespace](#)

AllObjects(T).GetEnumerator Method

Enumeration of all objects of type T

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<T> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of T)
```

C++

```
public:  
virtual IEnumerable<T>^ GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<'T>  
override GetEnumerator : unit -> IEnumerable<'T>
```

Return Value

Type: [IEnumerable\(T\)](#)

The enumerator of T

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[AllObjects\(T\)Class](#)

[VelocityDb Namespace](#)

AllObjects(T).AllObjectsIterator Class

Iterator of type [AllObjects\(T\)](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OfType.OfTypeIterator](#)

VelocityDb.AllObjects(T).AllObjectsIterator

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class AllObjectsIterator : OfType.OfTypeIterator,  
    IEnumerable<T>, IEnumerable
```

VB

```
Public Class AllObjectsIterator  
    Inherits OfType.OfTypeIterator  
    Implements IEnumerable(Of T), IEnumerable
```

C++


```
public ref class AllObjectsIterator : public OfType.OfTypeIterator,  
    IEnumerable<T>, IEnumerable
```

F#

```
type AllObjectsIterator =  
    class  
        inherit OfType.OfTypeIterator  
        interface IEnumerable<'T>  
        interface IEnumerable  
    end
```

The AllObjects(T).AllObjectsIterator generic type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------|
|  | GetEnumerator | Returns an enumerator that iterates through the collection. |


See Also

[VelocityDb Namespace](#)

AllObjectsIterator.AllObjectsIterator Methods

The [AllObjects\(T\).AllObjectsIterator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------|
|  | GetEnumerator | Returns an enumerator that iterates through the collection. |

See Also

[AllObjects\(T\).AllObjectsIterator Class](#)

[VelocityDb Namespace](#)

AllObjects(T).AllObjectsIterator.GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<T> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of T)
```

C++

```
public:  
virtual IEnumerator<T>^ GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<'T>  
override GetEnumerator : unit -> IEnumerator<'T>
```

Return Value

Type: [IEnumerator\(T\)](#)

An enumerator that can be used to iterate through the collection.

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[AllObjects\(T\).AllObjectsIterator Class](#)

[VelocityDb Namespace](#)

AllObjectsExtension Class

A few extensions to improve performance of Linq for Objects queries

Inheritance Hierarchy

[System.Object](#)

VelocityDb.AllObjectsExtension

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class AllObjectsExtension
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class AllObjectsExtension
```

C++







```
[ExtensionAttribute]  
public ref class AllObjectsExtension abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type AllObjectsExtension = class end
```

The **AllObjectsExtension** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Count(T) | Override to improve performance over IEnumerable LINQ extension |
|   | ElementAt(T) | Override LINQ for faster access |
|   | Skip(TSource) | Bypasses a specified number of elements in a sequence and then returns the remaining elements. Overrides Skip(TSource)(IEnumerable(TSource), Int32) for improved performance |







See Also

[VelocityDb Namespace](#)

AllObjectsExtension.AllObjectsExtension Methods

The [AllObjectsExtension](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Count(T) | Override to improve performance over IEnumerable LINQ extension |
|   | ElementAt(T) | Override LINQ for faster access |
|   | Skip(TSource) | Bypasses a specified number of elements in a sequence and then returns the remaining elements. Overrides Skip(TSource)(IEnumerable(TSource), Int32) for improved performance |

See Also

[AllObjectsExtension Class](#)

[VelocityDb Namespace](#)

AllObjectsExtension.Count(T) Method

Override to improve performance over IEnumerable LINQ extension

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int Count<T>(
    this AllObjects<T> sourceCollection
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Count(Of T) (
    sourceCollection As AllObjects(Of T)
) As Integer
```

C++

```
public:
    [ExtensionAttribute]
    generic<typename T>
    static int Count(
        AllObjects<T>^ sourceCollection
    )
```

F#

```
[<ExtensionAttribute>]
static member Count :
    sourceCollection : AllObjects<'T> -> int
```

Parameters

sourceCollection

Type: [VelocityDb.AllObjects\(T\)](#)

the collection

Type Parameters

T

key type

Return Value

Type: [Int32](#)

Size of the collection

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [AllObjects\(T\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[AllObjectsExtension Class](#)

[VelocityDb Namespace](#)

AllObjectsExtension.ElementAt(*T*) Method

Override LINQ for faster access

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static T ElementAt<T>(
    this AllObjects<T> source,
    int index
)
```

VB

```
<ExtensionAttribute>
Public Shared Function ElementAt(Of T) (
    source As AllObjects(Of T),
    index As Integer
) As T
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
static T ElementAt(
    AllObjects<T>^ source,
    int index
)
```

F#

```
[<ExtensionAttribute>]
static member ElementAt :
    source : AllObjects<'T> *
    index : int -> 'T
```

Parameters

source

Type: [VelocityDb.AllObjects\(*T*\)](#)

The source enumeration

index

Type: [System.Int32](#)

The index requested

Type Parameters

T

VelocityDB Class Library

The type of element requested

Return Value

Type: **T**

Element at requested index

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [AllObjects\(T\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[AllObjectsExtension Class](#)

[VelocityDb Namespace](#)

AllObjectsExtension.Skip(TSource) Method

Bypasses a specified number of elements in a sequence and then returns the remaining elements. Overrides [Skip\(TSource\)\(IEnumerable\(TSource\), Int32\)](#) for improved performance

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEnumerable<TSource> Skip<TSource>(
    this AllObjects<TSource> source,
    int count
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Skip(Of TSource) (
    source As AllObjects(Of TSource),
    count As Integer
) As IEnumerable(Of TSource)
```

C++

```
public:
[ExtensionAttribute]
generic<typename TSource>
static IEnumerable<TSource>^ Skip(
    AllObjects<TSource>^ source,
    int count
)
```

F#

```
[<ExtensionAttribute>]
static member Skip :
    source : AllObjects<'TSource> *
    count : int -> IEnumerable<'TSource>
```

Parameters

source

Type: [VelocityDb.AllObjects\(TSource\)](#)

Source collection

count

Type: [System.Int32](#)

The number of elements to skip before returning the remaining elements.

VelocityDB Class Library

Type Parameters

TSource

The type of the elements of source

Return Value

Type: [IEnumerable\(TSource\)](#)

An [IEnumerable\(T\)](#) that contains the elements that occur after the specified index in the input sequence.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [AllObjects\(TSource\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[AllObjectsExtension Class](#)

[VelocityDb Namespace](#)

AutoPlacement Class

This is the placement manager of objects persisted using [Persist\(Object\)](#), it is automatically created when the first object of a type is persisted. It is the first object in the database of each type mapped Database.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Placement](#)

VelocityDb.AutoPlacement

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class AutoPlacement : Placement
```

VB

```
<SerializableAttribute>
Public Class AutoPlacement
    Inherits Placement
```

C++




```
[SerializableAttribute]
public ref class AutoPlacement : public Placement
```

F#





```
[<SerializableAttribute>]
type AutoPlacement =
    class
        inherit Placement
    end
```

The **AutoPlacement** type exposes the following members.



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | We want to cache all AutoPlacement objects (small and frequently used) (Overrides OptimizedPersistable.Cache.) |
|  | Count | Get the total number of created objects of this type |
|  | Counter | Get the total number of created objects of this type |

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|  | IncrementTryDatabaseNumber | Moves to a new Database number used when placing (persisting) objects (Overrides Placement.IncrementTryDatabaseNumber() .) |
|  | InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase) .) |
|   | Open | Tries to open AutoPlacement object for a given database |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |




See Also

[VelocityDb Namespace](#)

AutoPlacement.AutoPlacement Properties

The [AutoPlacement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | We want to cache all AutoPlacement objects (small and frequently used) (Overrides OptimizedPersistable.Cache.) |
|  | Count | Get the total number of created objects of this type |
|  | Counter | Get the total number of created objects of this type |

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.Cache Property

We want to cache all AutoPlacement objects (small and frequently used)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.Count Property

Get the total number of created objects of this type

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Count { get; }
```

VB

```
Public ReadOnly Property Count As ULong  
    Get
```

C++

```
public:  
property unsigned long long Count {  
    unsigned long long get ();  
}
```

F#

```
member Count : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.Counter Property

Get the total number of created objects of this type

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Counter { get; set; }
```

VB

```
Public Property Counter As ULong  
    Get  
    Set
```

C++

```
public:  
property unsigned long long Counter {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
member Counter : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also





[AutoPlacement Class](#)

[VelocityDb Namespace](#)



AutoPlacement.AutoPlacement Methods

The [AutoPlacement](#) type exposes the following members.

Methods

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|  | IncrementTryDatabaseNumber | Moves to a new Database number used when placing (persisting) objects (Overrides Placement.IncrementTryDatabaseNumber() .) |
|  | InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase) .) |
|   | Open | Tries to open AutoPlacement object for a given database |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.IncrementTryDatabaseNumber Method

Moves to a new Database number used when placing (persisting) objects

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint IncrementTryDatabaseNumber ()
```

VB

```
Public Overrides Function IncrementTryDatabaseNumber As UInteger
```

C++

```
public:  
virtual unsigned int IncrementTryDatabaseNumber () override
```

F#

```
abstract IncrementTryDatabaseNumber : unit -> uint32  
override IncrementTryDatabaseNumber : unit -> uint32
```

Return Value

Type: [UInt32](#)

The Database number before incrementing

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.InitializeAfterRead Method

[Missing <summary> documentation for "M:VelocityDb.AutoPlacement.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.AutoPlacement.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

AutoPlacement.Open Method

Tries to open [AutoPlacement](#) object for a given database

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IOptimizedPersistable Open(  
    SessionBase session,  
    uint databaseNumber,  
    ushort objectsPerPage,  
    bool usesAutoIncrement  
)
```

VB

```
Public Shared Function Open (  
    session As SessionBase,  
    databaseNumber As UInteger,  
    objectsPerPage As UShort,  
    usesAutoIncrement As Boolean  
) As IOptimizedPersistable
```

C++

```
public:  
static IOptimizedPersistable^ Open(  
    SessionBase^ session,  
    unsigned int databaseNumber,  
    unsigned short objectsPerPage,  
    bool usesAutoIncrement  
)
```

F#

```
static member Open :  
    session : SessionBase *  
    databaseNumber : uint32 *  
    objectsPerPage : uint16 *  
    usesAutoIncrement : bool -> IOptimizedPersistable
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

Active session

databaseNumber

Type: [System.UInt32](#)

Database number used by [Type](#) using this [AutoPlacement](#)

objectsPerPage

Type: [System.UInt16](#)

Requested max number of objects per page

usesAutoIncrement

Type: [System.Boolean](#)

Type for which [AutoPlacement](#) is used must not use any auto increment fields or else parallel add of this object type is not possible with simple persist.

Return Value

Type: [IOptimizedPersistable](#)

`null` if doesn't exist, the persistent [AutoPlacement](#) object if we able to read it from the [Database](#) without a lock conflict, a transient [AutoPlacement](#) if auto increment isn't used and we get a lock conflict; otherwise throw lock conflict exception

See Also

[AutoPlacement Class](#)

[VelocityDb Namespace](#)

CacheEnum Enumeration

Choices for object caching

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum CacheEnum
```

VB

```
Public Enumeration CacheEnum
```

C++

```
public enum class CacheEnum
```

F#

```
type CacheEnum
```

Members

| Member name | Value | Description |
|-------------------------|-------|---------------------------------------------|
| No | 0 | Do not cache |
| Yes | 1 | Always cache |
| is64Bit | 2 | Cache if running as 64 bit process |
| RamAvailable5Gb | 3 | Cache if at least 5GB memory is available |
| RamAvailable10Gb | 4 | Cache if at least 10GB memory is available |
| RamAvailable20Gb | 5 | Cache if at least 20GB memory is available |
| Available100Gb | 6 | Cache if at least 100GB memory is available |

See Also

[VelocityDb Namespace](#)

Database Class

Represents a Database which corresponds to file system file.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Database

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class Database : OptimizedPersistable, IEnumerable<Page>, IEnumerable, IEqualityComparer<Database>, IDisposable
```

VB

```
<SerializableAttribute>  
Public Class Database  
    Inherits OptimizedPersistable  
    Implements IEnumerable(Of Page), IEnumerable, IEqualityComparer(Of Database), IDisposable
```

C++
















```
[SerializableAttribute]  
public ref class Database : public OptimizedPersistable, IEnumerable<Page^>, IEnumerable, IEqualityComparer<Database^>, IDisposable
```

F#





```
[<SerializableAttribute>]  
type Database =  
    class  
        inherit OptimizedPersistable  
        interface IEnumerable<Page>  
        interface IEnumerable  
        interface IEqualityComparer<Database>  
        interface IDisposable  
    end
```











The **Database** type exposes the following members.

Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | We want to be sure that the object representing a Database can share its page with other objects so override <code>AllowOtherTypesOnSamePage</code> and return true. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage .) |
|  CachedVerified | Setting this property to false will trigger a comparison of the cached version of the Database with the on disk version. Each new transaction also sets this property to false for all cached databases. You may alternatively use the session API <code>ForceDatabaseCacheValidation()</code> to trigger validation of all cached databases. Setting it to false when the Database has been updated in the current transaction is ignored. |
|  FileInfo | FileInfo for this Database , not set for <code>ServerClientSession</code> |
|  FileStream | Stream used for reading and writing this Database |
|  HighestPageNumber | Indicates the highest <code>tryPageNumber</code> number currently in use for this database. |
|  IsDeleted | Gets info about this Database about to be deleted or not. |
|  IsLocal | Gets info about this Database being on the local host or not. |
|  IsNew | Indicates if database was created in the current <code>transactionNumber</code> |
|  IsUpdated | Gets the updated state of the object (Overrides OptimizedPersistable.IsUpdated .) |
|  Location | The <code>DatabaseLocation</code> of this Database |
|  Name | Optional name associated with a Database |
|  NumberOfPages | Indicates how many pages this database contains. |
|  ObjectCachingDefaultPolicy | Default strategy for this Database , defaults to ObjectCachingDefaultPolicy . (strategy is not persisted) |
|  PageCacheEnabled | Turns on or off Page caching for this Database |
|  PageOffset | |

Methods



| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllObjects(T) | Gets an object used for enumerating all objects in this Database |
|  Bytes | Avoid using this one for now, internal use. |
|  CachedPage | Try to retrieve a cached Page |
|  CloneAs | Any Database that only uses <code>OidShort</code> references can be cloned without changing any of the internals of the Database. This function simply copies the |

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Database file. It is the user's responsibility to know that only OidShort references are used within the Database or else such references will still keep the original Database number. |
|  | Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |
|  | Equals(Database) | Compares by version and page 0 offset |
|  | Equals(Database, Database) | Determines whether the specified objects are equal. |
|  | GetEnumerator | Gets an enumerable sequence of all the pages in this database. |
|  | GetHashCode | Returns a hash code for the specified object. |
|  | OfType | Gets an object used for enumerating all objects in this Database."/> |
|  | Pages | Gets an enumerable sequence of all the pages in this database. |
|  | ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion, Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  | ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString().) |
|  | WritePageBytes | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------|-------------|
|  | InitialReservedDatabaseNumbers | |
|  | m_guid | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
















See Also

[VelocityDb Namespace](#)

Database.Database Properties

The [Database](#) type exposes the following members.

Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | We want to be sure that the object representing a Database can share its page with other objects so override AllowOtherTypesOnSamePage and return true. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage .) |
|  CachedVerified | Setting this property to false will trigger a comparison of the cached version of the Database with the on disk version. Each new transaction also sets this property to false for all cached databases. You may alternatively use the session API ForceDatabaseCacheValidation() to trigger validation of all cached databases. Setting it to false when the Database has been updated in the current transaction is ignored. |
|  FileInfo | FileInfo for this Database , not set for ServerClientSession |
|  FileStream | Stream used for reading and writing this Database |
|  HighestPageNumber | Indicates the highest tryPageNumber number currently in use for this database. |
|  IsDeleted | Gets info about this Database about to be deleted or not. |
|  IsLocal | Gets info about this Database being on the local host or not. |
|  IsNew | Indicates if database was created in the current transactionNumber |
|  IsUpdated | Gets the updated state of the object (Overrides OptimizedPersistable.IsUpdated .) |
|  Location | The DatabaseLocation of this Database |
|  Name | Optional name associated with a Database |
|  NumberOfPages | Indicates how many pages this database contains. |
|  ObjectCachingDefaultPolicy | Default strategy for this Database , defaults to ObjectCachingDefaultPolicy . (strategy is not persisted) |
|  PageCacheEnabled | Turns on or off Page caching for this Database |
|  PageOffset | |

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.AllowOtherTypesOnSamePage Property

We want to be sure that the object representing a Database can share its page with other objects so override AllowOtherTypesOnSamePage and return true.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.CachedVerified Property

Setting this property to false will trigger a comparison of the cached version of the Database with the on disk version. Each new transaction also sets this property to false for all cached databases. You may alternatively use the session API ForceDatabaseCacheValidation() to trigger validation of all cached databases. Setting it to false when the Database has been updated in the current transaction is ignored.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool CachedVerified { get; set; }
```

VB

```
Public Property CachedVerified As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool CachedVerified {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member CachedVerified : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.FileInfo Property

[FileInfo](#)

for this [Database](#), not set for ServerClientSession

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FileInfo FileInfo { get; set; }
```

VB

```
Public Property FileInfo As FileInfo  
    Get  
    Set
```

C++

```
public:  
property FileInfo^ FileInfo {  
    FileInfo^ get ();  
    void set (FileInfo^ value);  
}
```

F#

```
member FileInfo : FileInfo with get, set
```

Property Value

Type: [FileInfo](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.FileStream Property [Stream](#)

used for reading and writing this [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Stream FileStream { get; set; }
```

VB

```
Public Property FileStream As Stream  
    Get  
    Set
```

C++

```
public:  
property Stream^ FileStream {  
    Stream^ get ();  
    void set (Stream^ value);  
}
```

F#

```
member FileStream : Stream with get, set
```

Property Value

Type: [Stream](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.HighestPageNumber Property

Indicates the highest tryPageNumber number currently in use for this database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort HighestPageNumber { get; }
```

VB

```
Public ReadOnly Property HighestPageNumber As UShort  
    Get
```

C++

```
public:  
property unsigned short HighestPageNumber {  
    unsigned short get ();  
}
```

F#

```
member HighestPageNumber : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.IsDeleted Property

Gets info about this Database about to be deleted or not.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsDeleted { get; }
```

VB

```
Public ReadOnly Property IsDeleted As Boolean  
    Get
```

C++

```
public:  
property bool IsDeleted {  
    bool get ();  
}
```

F#

```
member IsDeleted : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.IsLocal Property

Gets info about this Database being on the local host or not.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsLocal { get; }
```

VB

```
Public ReadOnly Property IsLocal As Boolean  
    Get
```

C++

```
public:  
property bool IsLocal {  
    bool get ();  
}
```

F#

```
member IsLocal : bool with get
```

Return Value

Type: [Boolean](#)

`true` if this Database is located on the local host; otherwise, `false`.

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.IsNew Property

Indicates if database was created in the current transactionNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsNew { get; }
```

VB

```
Public ReadOnly Property IsNew As Boolean  
    Get
```

C++

```
public:  
property bool IsNew {  
    bool get ();  
}
```

F#

```
member IsNew : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.IsUpdated Property

Gets the updated state of the object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override bool IsUpdated { get; set; }
```

```
VB  
Public Overrides Property IsUpdated As Boolean  
    Get  
    Set
```

```
C++  
public:  
virtual property bool IsUpdated {  
    bool get () override;  
    void set (bool value) override;  
}
```

```
F#  
abstract IsUpdated : bool with get, set  
override IsUpdated : bool with get, set
```

Property Value

Type: [Boolean](#)

true if updated; otherwise false

Implements

[IOptimizedPersistable.IsUpdated](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Location Property

The DatabaseLocation of this Database

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation Location { get; }
```

VB

```
Public ReadOnly Property Location As DatabaseLocation  
    Get
```

C++

```
public:  
property DatabaseLocation^ Location {  
    DatabaseLocation^ get ();  
}
```

F#

```
member Location : DatabaseLocation with get
```

Property Value

Type: [DatabaseLocation](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Name Property

Optional name associated with a Database

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; set; }
```

VB

```
Public Property Name As String  
    Get  
    Set
```

C++

```
public:  
property String^ Name {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member Name : string with get, set
```

Property Value

Type: [String](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.NumberOfPages Property

Indicates how many pages this database contains.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort NumberOfPages { get; }
```

VB

```
Public ReadOnly Property NumberOfPages As UShort  
    Get
```

C++

```
public:  
property unsigned short NumberOfPages {  
    unsigned short get ();  
}
```

F#

```
member NumberOfPages : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.ObjectCachingDefaultPolicy Property

Default strategy for this [Database](#), defaults to [ObjectCachingDefaultPolicy](#). (strategy is not persisted)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public CacheEnum ObjectCachingDefaultPolicy { get; set; }
```

VB

```
Public Property ObjectCachingDefaultPolicy As CacheEnum  
    Get  
    Set
```

C++

```
public:  
property CacheEnum ObjectCachingDefaultPolicy {  
    CacheEnum get ();  
    void set (CacheEnum value);  
}
```

F#

```
member ObjectCachingDefaultPolicy : CacheEnum with get, set
```

Property Value

Type: [CacheEnum](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.PageCacheEnabled Property

Turns on or off [Page](#) caching for this [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool PageCacheEnabled { get; set; }
```

VB

```
Public Property PageCacheEnabled As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool PageCacheEnabled {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member PageCacheEnabled : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.PageOffset Property

[Missing <summary> documentation for "P:VelocityDb.Database.PageOffset"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageOffset PageOffset { get; }
```

VB

```
Public ReadOnly Property PageOffset As PageOffset  
    Get
```

C++

```
public:  
property PageOffset^ PageOffset {  
    PageOffset^ get ();  
}
```

F#

```
member PageOffset : PageOffset with get
```

Property Value

Type: [PageOffset](#)

See Also















[Database Class](#)

[VelocityDb Namespace](#)



Database.Database Methods

The [Database](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllObjects(T) | Gets an object used for enumerating all objects in this Database |
|  Bytes | Avoid using this one for now, internal use. |
|  CachedPage | Try to retrieve a cached Page |
|  CloneAs | Any Database that only uses OidShort references can be cloned without changing any of the internals of the Database. This function simply copies the Database file. It is the user's responsibility to know that only OidShort references are used within the Database or else such references will still keep the original Database number. |
|  Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |
|  Equals(Database) | Compares by version and page 0 offset |
|  Equals(Database, Database) | Determines whether the specified objects are equal. |
|  GetEnumerator | Gets an enumerable sequence of all the pages in this database. |
|  GetHashCode | Returns a hash code for the specified object. |
|  OfType | Gets an object used for enumerating all objects in this Database."/> |
|  Pages | Gets an enumerable sequence of all the pages in this database. |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString().) |
|  WritePageBytes | |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.AllObjects(T) Method

Gets an object used for enumerating all objects in this Database

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AllObjects<T> AllObjects<T>(
    bool includeSubClasses = true
)
```

VB

```
Public Function AllObjects(Of T) (
    Optional includeSubClasses As Boolean = true
) As AllObjects(Of T)
```

C++

```
public:
generic<typename T>
AllObjects<T>^ AllObjects(
    bool includeSubClasses = true
)
```

F#

```
member AllObjects :
    ?includeSubClasses : bool
(* Defaults:
    let_includeSubClasses = defaultArg includeSubClasses true
*)
-> AllObjects<'T>
```

Parameters

includeSubClasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes

Type Parameters

T

The type of object we are looking for

Return Value

Type: [AllObjects\(T\)](#)

The enumeration wrapper object

VelocityDB Class Library

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Bytes Method

Avoid using this one for now, internal use.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public byte[] Bytes ()
```

VB

```
Public Function Bytes As Byte ()
```

C++

```
public:  
array<unsigned char>^ Bytes ()
```

F#

```
member Bytes : unit -> byte[]
```

Return Value

Type: [Byte\[\]](#)

The bytes of the Database

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.CachedPage Method

Try to retrieve a cached [Page](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Page CachedPage (  
    ushort pageNumber  
)
```

VB

```
Public Function CachedPage (  
    pageNumber As UShort  
) As Page
```

C++

```
public:  
Page^ CachedPage (  
    unsigned short pageNumber  
)
```

F#

```
member CachedPage :  
    pageNumber : uint16 -> Page
```

Parameters

pageNumber

Type: [System.UInt16](#)

Page number of page to look for in cache

Return Value

Type: [Page](#)

[Page](#) if found in cache; otherwise null

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.CloneAs Method

Any Database that only uses OidShort references can be cloned without changing any of the internals of the Database. This function simply copies the Database file. It is the user's responsibility to know that only OidShort references are used within the Database or else such references will still keep the original Database number.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Database CloneAs(  
    uint otherAvaialbleDatabaseNumber  
)
```

VB

```
Public Function CloneAs (  
    otherAvaialbleDatabaseNumber As UInteger  
) As Database
```

C++

```
public:  
Database^ CloneAs(  
    unsigned int otherAvaialbleDatabaseNumber  
)
```

F#

```
member CloneAs :  
    otherAvaialbleDatabaseNumber : uint32 -> Database
```

Parameters

otherAvaialbleDatabaseNumber

Type: [System.UInt32](#)

The desired Id of the Database clone

Return Value

Type: [Database](#)

The Database clone

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)



See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Equals Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------|
|  | Equals(Database) | Compares by version and page 0 offset |
|  | Equals(Database, Database) | Determines whether the specified objects are equal. |

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Equals Method (Database)

Compares by version and page 0 offset

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    Database other  
)
```

VB

```
Public Function Equals (  
    other As Database  
) As Boolean
```

C++

```
public:  
bool Equals(  
    Database^ other  
)
```

F#

```
member Equals :  
    other : Database -> bool
```

Parameters

other

Type: [VelocityDb.Database](#)

Other [Database](#) object

Return Value

Type: [Boolean](#)

true if matching; otherwise false

See Also

[Database Class](#)

[Equals Overload](#)

[VelocityDb Namespace](#)

Database.Equals Method (Database, Database)

Determines whether the specified objects are equal.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    Database x,  
    Database y  
)
```

VB

```
Public Function Equals (  
    x As Database,  
    y As Database  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Database^ x,  
    Database^ y  
) sealed
```

F#

```
abstract Equals :  
    x : Database *  
    y : Database -> bool  
override Equals :  
    x : Database *  
    y : Database -> bool
```

Parameters

x

Type: [VelocityDb.Database](#)

The first object of type *T* to compare.

y

Type: [VelocityDb.Database](#)

The second object of type *T* to compare.

Return Value

Type: [Boolean](#)

`true` (True in Visual Basic) if the specified objects are equal; otherwise, `false` (False in Visual Basic).

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)

See Also

[Database Class](#)

[Equals Overload](#)

[VelocityDb Namespace](#)

Database.GetEnumerator Method

Gets an enumerable sequence of all the pages in this database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Page> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of Page)
```

C++

```
public:  
virtual IEnumerable<Page^> GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<Page>  
override GetEnumerator : unit -> IEnumerable<Page>
```

Return Value

Type: [IEnumerable\(Page\)](#)

Enumeration of pages in a Database

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.GetHashCode Method

Returns a hash code for the specified object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode(  
    Database db  
)
```

VB

```
Public Function GetHashCode (  
    db As Database  
) As Integer
```

C++

```
public:  
virtual int GetHashCode(  
    Database^ db  
) sealed
```

F#

```
abstract GetHashCode :  
    db : Database -> int  
override GetHashCode :  
    db : Database -> int
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDb.Database.GetHashCode(VelocityDb.Database)"]

Return Value

Type: [Int32](#)

A hash code for the specified object.

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------|
| ArgumentNullException | The type of <i>obj</i> is a reference type and <i>obj</i> is a null reference (<i>Nothing</i> in Visual Basic). |

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.OfType Method

Gets an object used for enumerating all objects in this Database."/>

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OfType OfType(  
    Type type,  
    bool includeSubClasses = true  
)
```

VB

```
Public Function OfType (  
    type As Type,  
    Optional includeSubClasses As Boolean = true  
) As OfType
```

C++

```
public:  
OfType^ OfType(  
    Type^ type,  
    bool includeSubClasses = true  
)
```

F#

```
member OfType :  
    type : Type *  
    ?includeSubClasses : bool  
(* Defaults:  
    let _includeSubClasses = defaultArg includeSubClasses true  
)  
-> OfType
```

Parameters

type

Type: [System.Type](#)

Type to look for

includeSubClasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes

Return Value

Type: [OfType](#)

The enumeration wrapper **OfType(Type, Boolean)**object

VelocityDB Class Library

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Pages Method

Gets an enumerable sequence of all the pages in this database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Page> Pages (  
    bool stubsOnly = true  
)
```

VB

```
Public Function Pages (  
    Optional stubsOnly As Boolean = true  
) As IEnumerable(Of Page)
```

C++

```
public:  
IEnumerable<Page^>^ Pages (  
    bool stubsOnly = true  
)
```

F#

```
member Pages :  
    ?stubsOnly : bool  
(* Defaults:  
    let_stubOnly = defaultArg stubsOnly true  
)  
-> IEnumerable<Page>
```

Parameters

stubsOnly (Optional)

Type: [System.Boolean](#)

Load page stub only?

Return Value

Type: [IEnumerable\(Page\)](#)

Enumeration of pages in a Database

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
    Page^ page,  
    bool useOidShort,  
    Schema^ schema,  
    bool openRefs,  
    List<IOptimizedPersistable^>^ toLoadMembers,  
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,Vel

ocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.Database.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.ToString Method

Displays class name plus object id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.WritePageBytes Method

[Missing <summary> documentation for

"M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void WritePageBytes (
    byte[] bytes,
    Page page,
    Dictionary<ushort, long> dbPageWrites,
    ulong transaction,
    Stream fStream,
    byte[] pageInfoBytes
)
```

VB

```
Public Sub WritePageBytes (
    bytes As Byte(),
    page As Page,
    dbPageWrites As Dictionary(Of UShort, Long),
    transaction As ULong,
    fStream As Stream,
    pageInfoBytes As Byte()
)
```

C++

```
public:
void WritePageBytes (
    array<unsigned char>^ bytes,
    Page^ page,
    Dictionary<unsigned short, long long>^ dbPageWrites,
    unsigned long long transaction,
    Stream^ fStream,
    array<unsigned char>^ pageInfoBytes
)
```

F#

```
member WritePageBytes :
    bytes : byte[] *
    page : Page *
    dbPageWrites : Dictionary<uint16, int64> *
    transaction : uint64 *
    fStream : Stream *
    pageInfoBytes : byte[] -> unit
```

Parameters

bytes

Type: [System.Byte\[\]](#)

[Missing <param name="bytes"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

dbPageWrites

Type: [System.Collections.Generic.Dictionary{UInt16, Int64}](#)

[Missing <param name="dbPageWrites"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

transaction

Type: [System.UInt64](#)

[Missing <param name="transaction"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

fStream

Type: [System.IO.Stream](#)

[Missing <param name="fStream"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

pageInfoBytes

Type: [System.Byte\[\]](#)

[Missing <param name="pageInfoBytes"/> documentation for "M:VelocityDb.Database.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.Database Fields

The [Database](#) type exposes the following members.

Fields

| | Name | Description |
|--------------------------------------------------------------------------------------------|------------------------------------------------|-------------|
|  S | InitialReservedDatabaseNumbers | |
|  | m_guid | |

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.InitialReservedDatabaseNumbers Field

[Missing <summary> documentation for "F:VelocityDb.Database.InitialReservedDatabaseNumbers"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint InitialReservedDatabaseNumbers = 9
```

VB

```
Public Const InitialReservedDatabaseNumbers As UInteger = 9
```

C++

```
public:  
literal unsigned int InitialReservedDatabaseNumbers = 9
```

F#

```
static val mutable InitialReservedDatabaseNumbers: uint32
```

Field Value

Type: [UInt32](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

Database.m_guid Field

[Missing <summary> documentation for "F:VelocityDb.Database.m_guid"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[NonSerializedAttribute]  
public Guid m_guid
```

VB

```
<NonSerializedAttribute>  
Public m_guid As Guid
```

C++

```
public:  
[NonSerializedAttribute]  
Guid m_guid
```

F#

```
[<NonSerializedAttribute>]  
val mutable m_guid: Guid
```

Field Value

Type: [Guid](#)

See Also

[Database Class](#)

[VelocityDb Namespace](#)

DatabaseLocation Class

Represents a host and a directory in which a range of databases are stored. The range is a database number range.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.DatabaseLocation

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class DatabaseLocation : OptimizedPersistable,
    IComparable
```

VB

```
<SerializableAttribute>
Public Class DatabaseLocation
    Inherits OptimizedPersistable
    Implements IComparable
```

C++


```
[SerializableAttribute]
public ref class DatabaseLocation : public OptimizedPersistable,
    IComparable
```


F#

```
[<SerializableAttribute>]
type DatabaseLocation =
    class
        inherit OptimizedPersistable
        interface IComparable
    end
```



















The **DatabaseLocation** type exposes the following members.

Constructors










| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | DatabaseLocation(String, String, UInt32, UInt32, SessionBase) | Creates a transient Database location. A transient DatabaseLocation is made persistent by NewLocation(DatabaseLocation) |

| | | |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | DatabaseLocation(String, String, UInt32, UInt32, SessionBase, PageInfo.compressionKind, PageInfo.encryptionKind, Boolean, DatabaseLocation) | Creates a transient Database location. A transient DatabaseLocation is made persistent by NewLocation(DatabaseLocation) |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|



Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  BackupOfOrForLocation | Gets a linked DatabaseLocation that is either backup of location or is the location being backup up which is decided by IsBackupLocation |
|  Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache.) |
|  CompressPages | Gets page compression info |
|  DesKey | Gets the Des encryption key from a .des file in user Documents folder. (Environment.SpecialFolder.MyDocuments) |
|  DirectoryPath | Gets the full path to the directory hosting databases |
|  EndDatabaseNumber | Gets or sets the last database number within the range of databases for this location |
|  HostName | Gets the host name of this database location |
|  IsBackupLocation | Tells if this is a location used for backing up databases. |
|  IsDefaultLocation | Is this the default DatabaseLocation where Database 0, 1, 2, 3, 4 may be stored? |
|  IsLocal | Gets info about this location being on the local host or not. |
|  PageEncryption | Gets the encryption kind attribute. |
|  PlacementDatabaseNumber | Hint about where to persist DatabaseLocation (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  Replicas | |
|  RsaPrivateXmlKey | The rsaPrivateXmlKey data field is not persisted but can be set and read with this property |
|  RsaPublicXmlKey | The rsaPublicXmlKey data field is not persisted but can be set and read with this property |
|  RsaSignature | The rsaSignature data field is not persisted but can be set and read with this property |
|  StartDatabaseNumber | Gets and sets the start of the range of Database numbers for this location. |

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CompareTo | The default compare of two DatabaseLocation is by host name and directory (Overrides OptimizedPersistable.CompareTo(Object).) |
|  ContainsDatabase | Check if this location contains a particular Database |
|  ContainsNewDatabase | Check if this location contains a particular new database (not yet committed) Database |
|  DatabasePath | Get the full path to a Database |
|  Databases | Enumerates all Databases of this location |
|  InitializeAfterRead | Sets the desKey field for desEncrypted locations by reading .des file in Environment.SpecialFolder.MyDocuments (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)).) |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |

Extension Methods



| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb Namespace](#)

DatabaseLocation Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | DatabaseLocation(String, String, UInt32, UInt32, SessionBase) | Creates a transient Database location. A transient DatabaseLocation is made persistent by NewLocation(DatabaseLocation) |
|  | DatabaseLocation(String, String, UInt32, UInt32, SessionBase, PageInfo.compressionKind, PageInfo.encryptionKind, Boolean, DatabaseLocation) | Creates a transient Database location. A transient DatabaseLocation is made persistent by NewLocation(DatabaseLocation) |

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation Constructor (String, String, UInt32, UInt32, SessionBase)

Creates a transient Database location. A transient DatabaseLocation is made persistent by [NewLocation\(DatabaseLocation\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation(  
    string inHost,  
    string dir,  
    uint dbStartNum,  
    uint dbEndNum,  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    inHost As String,  
    dir As String,  
    dbStartNum As UInteger,  
    dbEndNum As UInteger,  
    session As SessionBase  
)
```

C++

```
public:  
DatabaseLocation(  
    String^ inHost,  
    String^ dir,  
    unsigned int dbStartNum,  
    unsigned int dbEndNum,  
    SessionBase^ session  
)
```

F#

```
new :  
    inHost : string *  
    dir : string *  
    dbStartNum : uint32 *  
    dbEndNum : uint32 *  
    session : SessionBase -> DatabaseLocation
```

Parameters

inHost

Type: [System.String](#)

VelocityDB Class Library

Hostname of the computer with the DatabaseLocation directory.

dir

Type: [System.String](#)

The full path of the directory

dbStartNum

Type: [System.UInt32](#)

The first Database Id number for this location

dbEndNum

Type: [System.UInt32](#)

The last Database Id number for this location.

session

Type: [VelocityDb.Session.SessionBase](#)

The active session used for creating the location.

See Also

[DatabaseLocation Class](#)

[DatabaseLocation Overload](#)

[VelocityDb Namespace](#)

DatabaseLocation Constructor (String, String, UInt32, UInt32, SessionBase, PageInfo.compressionKind, PageInfo.encryptedKind, Boolean, DatabaseLocation)

Creates a transient Database location. A transient DatabaseLocation is made persistent by [NewLocation\(DatabaseLocation\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation(  
    string inHost,  
    string dir,  
    uint dbStartNum,  
    uint dbEndNum,  
    SessionBase session,  
    PageInfo.compressionKind compressPages,  
    PageInfo.encryptedKind pageEncryption =  
PageInfo.encryptedKind.noEncryption,  
    bool isAbackupLocation = false,  
    DatabaseLocation backupOfOrForLocation = null  
)
```

VB

```
Public Sub New (  
    inHost As String,  
    dir As String,  
    dbStartNum As UInteger,  
    dbEndNum As UInteger,  
    session As SessionBase,  
    compressPages As PageInfo.compressionKind,  
    Optional pageEncryption As PageInfo.encryptedKind =  
PageInfo.encryptedKind.noEncryption,  
    Optional isAbackupLocation As Boolean = false,  
    Optional backupOfOrForLocation As DatabaseLocation = Nothing  
)
```

C++

```
public:  
DatabaseLocation(  
    String^ inHost,  
    String^ dir,  
    unsigned int dbStartNum,  
    unsigned int dbEndNum,  
    SessionBase^ session,  
    PageInfo.compressionKind compressPages,  
    PageInfo.encryptedKind pageEncryption =  
PageInfo.encryptedKind::noEncryption,
```

```
bool isAbackupLocation = false,  
DatabaseLocation^ backupOfOrForLocation = nullptr  
)
```

F#

```
new :  
    inHost : string *  
    dir : string *  
    dbStartNum : uint32 *  
    dbEndNum : uint32 *  
    session : SessionBase *  
    compressPages : PageInfo.compressionKind *  
    ?pageEncryption : PageInfo.encryptionKind *  
    ?isAbackupLocation : bool *  
    ?backupOfOrForLocation : DatabaseLocation  
(* Defaults:  
    let_pageEncryption = defaultArg pageEncryption  
PageInfo.encryptionKind.noEncryption  
    let_isAbackupLocation = defaultArg isAbackupLocation false  
    let_backupOfOrForLocation = defaultArg backupOfOrForLocation null  
*)  
-> DatabaseLocation
```

Parameters

inHost

Type: [System.String](#)

Hostname of the computer with the DatabaseLocation directory.

dir

Type: [System.String](#)

The full path of the directory

dbStartNum

Type: [System.UInt32](#)

The first Database Id number for this location

dbEndNum

Type: [System.UInt32](#)

The last Database Id number for this location.

session

Type: [VelocityDb.Session.SessionBase](#)

The active session used for creating the location.

compressPages

Type: [VelocityDb.PageInfo.compressionKind](#)

Compress Database pages in this location?

pageEncryption (Optional)

Type: [VelocityDb.PageInfo.encryptionKind](#)

VelocityDB Class Library

Specify what kind of Page encryption to use (if any)

isAbackupLocation (Optional)

Type: [System.Boolean](#)

Is this location a backup location for another location?

backupOfOrForLocation (Optional)

Type: [VelocityDb.DatabaseLocation](#)

The other location being backed up or the location contain the backups

See Also

[DatabaseLocation Class](#)



















[DatabaseLocation Overload](#)

[VelocityDb Namespace](#)

DatabaseLocation.DatabaseLocation Properties

The [DatabaseLocation](#) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  BackupOfOrForLocation | Gets a linked DatabaseLocation that is either backup of location or is the location being backup up which is decided by IsBackupLocation |
|  Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache.) |
|  CompressPages | Gets page compression info |
|  DesKey | Gets the Des encryption key from a .des file in user Documents folder. (Environment.SpecialFolder.MyDocuments) |
|  DirectoryPath | Gets the full path to the directory hosting databases |
|  EndDatabaseNumber | Gets or sets the last database number within the range of databases for this location |
|  HostName | Gets the host name of this database location |
|  IsBackupLocation | Tells if this is a location used for backing up databases. |
|  IsDefaultLocation | Is this the default DatabaseLocation where Database 0, 1, 2, 3, 4 may be stored? |
|  IsLocal | Gets info about this location being on the local host or not. |
|  PageEncryption | Gets the encryption kind attribute. |
|  PlacementDatabaseNumber | Hint about where to persist DatabaseLocation (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  Replicas | |
|  RsaPrivateXmlKey | The rsaPrivateXmlKey data field is not persisted but can be set and read with this property |
|  RsaPublicXmlKey | The rsaPublicXmlKey data field is not persisted but can be set and read with this property |
|  RsaSignature | The rsaSignature data field is not persisted but can be set and read with this property |
|  StartDatabaseNumber | Gets and sets the start of the range of Database numbers for this location. |

See Also

[DatabaseLocation Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

DatabaseLocation.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.BackupOfOrForLocation Property

Gets a linked DatabaseLocation that is either backup of location or is the location being backup up which is decided by [IsBackupLocation](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation BackupOfOrForLocation { get; set; }
```

VB

```
Public Property BackupOfOrForLocation As DatabaseLocation  
    Get  
    Set
```

C++

```
public:  
property DatabaseLocation^ BackupOfOrForLocation {  
    DatabaseLocation^ get ();  
    void set (DatabaseLocation^ value);  
}
```

F#

```
member BackupOfOrForLocation : DatabaseLocation with get, set
```

Property Value

Type: [DatabaseLocation](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.Cache Property

We do want to cache this kind of object since they are likely to be frequently used.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.CompressPages Property

Gets page compression info

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageInfo.compressionKind CompressPages { get; set; }
```

VB

```
Public Property CompressPages As PageInfo.compressionKind  
    Get  
    Set
```

C++

```
public:  
property PageInfo.compressionKind CompressPages {  
    PageInfo.compressionKind get ();  
    void set (PageInfo.compressionKind value);  
}
```

F#

```
member CompressPages : PageInfo.compressionKind with get, set
```

Return Value

Type: [PageInfo.compressionKind](#)

true if the databases in this location uses compression; otherwise, false.

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.DesKey Property

Gets the Des encryption key from a .des file in user Documents folder.
(Environment.SpecialFolder.MyDocuments)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public byte[] DesKey { get; set; }
```

VB

```
Public Property DesKey As Byte ()  
    Get  
    Set
```

C++

```
public:  
property array<unsigned char>^ DesKey {  
    array<unsigned char>^ get ();  
    void set (array<unsigned char>^ value);  
}
```

F#

```
member DesKey : byte[] with get, set
```

Property Value

Type: [Byte\[\]](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.DirectoryPath Property

Gets the full path to the directory hosting databases

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string DirectoryPath { get; }
```

VB

```
Public ReadOnly Property DirectoryPath As String  
    Get
```

C++

```
public:  
property String^ DirectoryPath {  
    String^ get ();  
}
```

F#

```
member DirectoryPath : string with get
```

Property Value

Type: [String](#)

The path as a string

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.EndDatabaseNumber Property

Gets or sets the last database number within the range of databases for this location

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint EndDatabaseNumber { get; set; }
```

VB

```
Public Property EndDatabaseNumber As UInteger  
    Get  
    Set
```

C++

```
public:  
property unsigned int EndDatabaseNumber {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
member EndDatabaseNumber : uint32 with get, set
```

Property Value

Type: [UInt32](#)

The last database number within the range of databases for this location

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.HostName Property

Gets the host name of this database location

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string HostName { get; }
```

VB

```
Public ReadOnly Property HostName As String  
    Get
```

C++

```
public:  
property String^ HostName {  
    String^ get ();  
}
```

F#

```
member HostName : string with get
```

Property Value

Type: [String](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.IsBackupLocation Property

Tells if this is a location used for backing up databases.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsBackupLocation { get; set; }
```

VB

```
Public Property IsBackupLocation As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool IsBackupLocation {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member IsBackupLocation : bool with get, set
```

Property Value

Type: [Boolean](#)

true if it is this location is used for backups; otherwise false.

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.IsDefaultLocation Property

Is this the default [DatabaseLocation](#) where [Database](#) 0, 1, 2, 3, 4 may be stored?

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsDefaultLocation { get; }
```

VB

```
Public ReadOnly Property IsDefaultLocation As Boolean  
    Get
```

C++

```
public:  
property bool IsDefaultLocation {  
    bool get ();  
}
```

F#

```
member IsDefaultLocation : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.IsLocal Property

Gets info about this location being on the local host or not.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsLocal { get; }
```

VB

```
Public ReadOnly Property IsLocal As Boolean  
    Get
```

C++

```
public:  
property bool IsLocal {  
    bool get ();  
}
```

F#

```
member IsLocal : bool with get
```

Return Value

Type: [Boolean](#)

true if the location is located on the local host; otherwise, false.

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.PageEncryption Property

Gets the encryption kind attribute.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageInfo.encryptionKind PageEncryption { get; set; }
```

VB

```
Public Property PageEncryption As PageInfo.encryptionKind  
    Get  
    Set
```

C++

```
public:  
property PageInfo.encryptionKind PageEncryption {  
    PageInfo.encryptionKind get ();  
    void set (PageInfo.encryptionKind value);  
}
```

F#

```
member PageEncryption : PageInfo.encryptionKind with get, set
```

Property Value

Type: [PageInfo.encryptionKind](#)

The [PageInfo.encryptionKind](#) used in this location

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.PlacementDatabaseNumber Property

Hint about where to persist DatabaseLocation

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint PlacementDatabaseNumber { get; }
```

VB

```
Public Overrides ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int PlacementDatabaseNumber {  
    unsigned int get () override;  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get  
override PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.PlacementDatabaseNumber](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.Replicas Property

[Missing <summary> documentation for "P:VelocityDb.DatabaseLocation.Replicas"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<string> Replicas { get; }
```

VB

```
Public ReadOnly Property Replicas As List(Of String)  
    Get
```

C++

```
public:  
property List<String^>^ Replicas {  
    List<String^>^ get ();  
}
```

F#

```
member Replicas : List<string> with get
```

Property Value

Type: [List\(String\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.RsaPrivateXmlKey Property

The rsaPrivateXmlKey data field is not persisted but can be set and read with this property

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string RsaPrivateXmlKey { get; set; }
```

VB

```
Public Property RsaPrivateXmlKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ RsaPrivateXmlKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member RsaPrivateXmlKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.RsaPublicXmlKey Property

The rsaPublicXmlKey data field is not persisted but can be set and read with this property

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string RsaPublicXmlKey { get; set; }
```

VB

```
Public Property RsaPublicXmlKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ RsaPublicXmlKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member RsaPublicXmlKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.RsaSignature Property

The rsaSignature data field is not persisted but can be set and read with this property

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public byte[] RsaSignature { get; set; }
```

VB

```
Public Property RsaSignature As Byte()  
    Get  
    Set
```

C++

```
public:  
property array<unsigned char>^ RsaSignature {  
    array<unsigned char>^ get ();  
    void set (array<unsigned char>^ value);  
}
```

F#

```
member RsaSignature : byte[] with get, set
```

Property Value

Type: [Byte\[\]](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.StartDatabaseNumber Property

Gets and sets the start of the range of [Database](#) numbers for this location.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint StartDatabaseNumber { get; set; }
```

VB

```
Public Property StartDatabaseNumber As UInteger  
    Get  
    Set
```

C++

```
public:  
property unsigned int StartDatabaseNumber {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
member StartDatabaseNumber : uint32 with get, set
```

Property Value

Type: [UInt32](#)

The first database number within the range of databases for this location

See Also










[DatabaseLocation Class](#)

[VelocityDb Namespace](#)



DatabaseLocation.DatabaseLocation Methods

The [DatabaseLocation](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CompareTo | The default compare of two DatabaseLocation is by host name and directory (Overrides OptimizedPersistable.CompareTo(Object).) |
|  ContainsDatabase | Check if this location contains a particular Database |
|  ContainsNewDatabase | Check if this location contains a particular new database (not yet committed) Database |
|  DatabasePath | Get the full path to a Database |
|  Databases | Enumerates all Databases of this location |
|  InitializeAfterRead | Sets the desKey field for desEncrypted locations by reading .des file in Environment.SpecialFolder.MyDocuments (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)).) |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.CompareTo Method

The default compare of two DatabaseLocation is by host name and directory

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int CompareTo(  
    Object obj  
)
```

VB

```
Public Overrides Function CompareTo (  
    obj As Object  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Object^ obj  
) override
```

F#

```
abstract CompareTo :  
    obj : Object -> int  
override CompareTo :  
    obj : Object -> int
```

Parameters

obj

Type: [System.Object](#)

The location to compare with

Return Value

Type: [Int32](#)

-1 if this location is less than the other, 0 if equal, otherwise 1

Implements

[IComparable.CompareTo\(Object\)](#)

[IComparable.CompareTo\(Object\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.ContainsDatabase Method

Check if this location contains a particular [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ContainsDatabase(  
    uint dbNum,  
    string extension = ".odb"  
)
```

VB

```
Public Function ContainsDatabase (  
    dbNum As UInteger,  
    Optional extension As String = ".odb"  
) As Boolean
```

C++

```
public:  
bool ContainsDatabase(  
    unsigned int dbNum,  
    String^ extension = L".odb"  
)
```

F#

```
member ContainsDatabase :  
    dbNum : uint32 *  
    ?extension : string  
(* Defaults:  
    let _extension = defaultArg extension ".odb"  
)  
-> bool
```

Parameters

dbNum

Type: [System.UInt32](#)

The database number to look for

extension (Optional)

Type: [System.String](#)

File name extension/id

Return Value

Type: [Boolean](#)

`true` if the location contains this database; otherwise, `false`.

VelocityDB Class Library

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.ContainsNewDatabase Method

Check if this location contains a particular new database (not yet committed) [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ContainsNewDatabase(  
    uint dbNum  
)
```

VB

```
Public Function ContainsNewDatabase (  
    dbNum As UInteger  
) As Boolean
```

C++

```
public:  
bool ContainsNewDatabase(  
    unsigned int dbNum  
)
```

F#

```
member ContainsNewDatabase :  
    dbNum : uint32 -> bool
```

Parameters

dbNum

Type: [System.UInt32](#)

The database number to look for

Return Value

Type: [Boolean](#)

true if the location contains this new uncommitted database; otherwise, false.

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.DatabasePath Method

Get the full path to a [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string DatabasePath(  
    uint dbNum  
)
```

VB

```
Public Function DatabasePath (  
    dbNum As UInteger  
) As String
```

C++

```
public:  
String^ DatabasePath(  
    unsigned int dbNum  
)
```

F#

```
member DatabasePath :  
    dbNum : uint32 -> string
```

Parameters

dbNum

Type: [System.UInt32](#)

The database number to look for

Return Value

Type: [String](#)

A [String](#) containing the full [Database](#) path.

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.Databases Method

Enumerates all Databases of this location

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Database> Databases ()
```

VB

```
Public Function Databases As IEnumerable (Of Database)
```

C++

```
public:  
IEnumerable<Database^>^ Databases ()
```

F#

```
member Databases : unit -> IEnumerable<Database>
```

Return Value

Type: [IEnumerable\(Database\)](#)

[Missing <returns> documentation for "M:VelocityDb.DatabaseLocation.Databases"]

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.InitializeAfterRead Method

Sets the desKey field for desEncrypted locations by reading .des file in Environment.SpecialFolder.MyDocuments

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.Persist Method

Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _persistRefs = defaultArg persistRefs true
    let _disableFlush = defaultArg disableFlush false
    let _toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
```

```
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of this persistent object

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue<IOptimizedPersistable>\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocation.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
Page^ page,  
bool useOidShort,  
Schema^ schema,  
bool openRefs,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int

32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.DatabaseLocation.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[DatabaseLocation Class](#)

[VelocityDb Namespace](#)

DatabaseLocations Class

Maintains collections of all existing [DatabaseLocations](#).

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.DatabaseLocations

Namespace: [VelocityDb](#)


Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

| C# |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>[SerializableAttribute] public class DatabaseLocations : OptimizedPersistable, IEnumerable<DatabaseLocation>, IEnumerable</pre> |
| VB |
| <pre><SerializableAttribute> Public Class DatabaseLocations Inherits OptimizedPersistable Implements IEnumerable(Of DatabaseLocation), IEnumerable</pre> |
| C++ |
| <pre>[SerializableAttribute] public ref class DatabaseLocations : public OptimizedPersistable, IEnumerable<DatabaseLocation^>, IEnumerable</pre> |
| F# |
| <pre>[<SerializableAttribute>] type DatabaseLocations = class inherit OptimizedPersistable interface IEnumerable<DatabaseLocation> interface IEnumerable end</pre> |







The **DatabaseLocations** type exposes the following members.

Properties



| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other |

| | |
|--|-------------------------------------------------------------------------------------|
| | types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|--|-------------------------------------------------------------------------------------|

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Default | Returns the DatabaseLocation of the startup/system Databases. |
|  GetEnumerator | Enumerates all DatabaseLocations |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  LocateDb | Returns the DatabaseLocation for which a Database exists in. |
|  LocationForDb | Returns the DatabaseLocation for which a Database belongs to. |
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |


See Also

[VelocityDb Namespace](#)

DatabaseLocations.DatabaseLocations Properties

The [DatabaseLocations](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage .) |

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also







[DatabaseLocations Class](#)

[VelocityDb Namespace](#)



DatabaseLocations.DatabaseLocations Methods

The [DatabaseLocations](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Default | Returns the DatabaseLocation of the startup/system Databases. |
|  GetEnumerator | Enumerates all DatabaseLocations |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase) .) |
|  LocateDb | Returns the DatabaseLocation for which a Database exists in. |
|  LocationForDb | Returns the DatabaseLocation for which a Database belongs to. |
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.Default Method

Returns the [DatabaseLocation](#) of the startup/system Databases.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation Default ()
```

VB

```
Public Function Default As DatabaseLocation
```

C++

```
public:  
DatabaseLocation^ Default ()
```

F#

```
member Default : unit -> DatabaseLocation
```

Return Value

Type: [DatabaseLocation](#)

The default [DatabaseLocation](#)

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.GetEnumerator Method

Enumerates all [DatabaseLocations](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<DatabaseLocation> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of DatabaseLocation)
```

C++

```
public:  
virtual IEnumerable<DatabaseLocation^> GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<DatabaseLocation>  
override GetEnumerator : unit -> IEnumerable<DatabaseLocation>
```

Return Value

Type: [IEnumerable\(DatabaseLocation\)](#)

An [IEnumerable\(T\)](#) object that can be used to iterate through the collection.

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.LocateDb Method

Returns the [DatabaseLocation](#) for which a Database exists in.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation LocateDb(  
    uint dbNum  
)
```

VB

```
Public Function LocateDb (  
    dbNum As UInteger  
) As DatabaseLocation
```

C++

```
public:  
DatabaseLocation^ LocateDb(  
    unsigned int dbNum  
)
```

F#

```
member LocateDb :  
    dbNum : uint32 -> DatabaseLocation
```

Parameters

dbNum

Type: [System.UInt32](#)

The Database number to lookup location for

Return Value

Type: [DatabaseLocation](#)

The looked up [DatabaseLocation](#) or null if the Database does not exist

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.LocationForDb Method

Returns the [DatabaseLocation](#) for which a Database belongs to.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation LocationForDb(  
    uint dbNum  
)
```

VB

```
Public Function LocationForDb (  
    dbNum As UInteger  
) As DatabaseLocation
```

C++

```
public:  
DatabaseLocation^ LocationForDb(  
    unsigned int dbNum  
)
```

F#

```
member LocationForDb :  
    dbNum : uint32 -> DatabaseLocation
```

Parameters

dbNum

Type: [System.UInt32](#)

The Database number to lookup location for

Return Value

Type: [DatabaseLocation](#)

The looked up [DatabaseLocation](#)

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DatabaseLocations.ReadMe Method

Used by code generator

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
```

```
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

memberBytes

Type: [System.Byte\[\]](#)

offset

Type: [System.Int32](#)

session

Type: [VelocityDb.Session.SessionBase](#)

page

Type: [VelocityDb.Page](#)

useOidShort

Type: [System.Boolean](#)

schema

Type: [VelocityDb.TypeInfo.Schema](#)

VelocityDB Class Library

openRefs

Type: [System.Boolean](#)

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

graphDepth

Type: [System.Int32](#)

graphDepthToLoad

Type: [System.Int32](#)

primitivesOnly

Type: [System.Boolean](#)

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[DatabaseLocations Class](#)

[VelocityDb Namespace](#)

DataCache Class

Object maintains a list of pages accessed by a session, pages are removed from the list when there is not enough memory available. This makes such pages eligible for garbage collection while in list pages are prevented from being garbage collected.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.DataCache

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class DataCache
```








```
VB
Public Class DataCache
```

```
C++
public ref class DataCache
```


```
F#
type DataCache = class end
```

The **DataCache** type exposes the following members.


Properties

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|   | AvailableRamMB | Current available memory in MB as calculated by using PerformanceCounter or current use relative to MaximumMemoryUse |
|   | MaximumMemoryUse | Maximum number of bytes to allow for this Process (reduces cached objects and pages when memory allocated reaches this soft limit) |
|   | MemoryUsedByThisProcess | The amount of virtual memory, in bytes, allocated for this process. |
|  | MinimumAvailableMegaBytes | Decide the minimum amount of available memory before reducing page caching and flushing updated pages. Only used by 32-bit process. |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | PageCacheEnabled | Enable or disable page cache. In cases where pages are not going to be revisited, performance may be improved by disabling the page cache. In other cases leave the page cache enabled (the default) |
|-----------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-----------------------------|
|  | Clear | Clear cache of cached pages |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | s_computerInfo | |









See Also

[VelocityDb Namespace](#)

DataCache.DataCache Properties

The [DataCache](#) type exposes the following members.

Properties

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AvailableRamMB | Current available memory in MB as calculated by using PerformanceCounter or current use relative to MaximumMemoryUse |
|   | MaximumMemoryUse | Maximum number of bytes to allow for this Process (reduces cached objects and pages when memory allocated reaches this soft limit) |
|   | MemoryUsedByThisProcess | The amount of virtual memory, in bytes, allocated for this process. |
|  | MinimumAvailableMegaBytes | Decide the minimum amount of available memory before reducing page caching and flushing updated pages. Only used by 32-bit process. |
|  | PageCacheEnabled | Enable or disable page cache. In cases where pages are not going to be revisited, performance may be improved by disabling the page cache. In other cases leave the page cache enabled (the default) |

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.AvailableRamMB Property

Current available memory in MB as calculated by using PerformanceCounter or current use relative to MaximumMemoryUse

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static uint AvailableRamMB { get; }
```

VB

```
Public Shared ReadOnly Property AvailableRamMB As UInteger  
    Get
```

C++

```
public:  
static property unsigned int AvailableRamMB {  
    unsigned int get ();  
}
```

F#

```
static member AvailableRamMB : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.MaximumMemoryUse Property

Maximum number of bytes to allow for this Process (reduces cached objects and pages when memory allocated reaches this soft limit)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static long MaximumMemoryUse { get; set; }
```

VB

```
Public Shared Property MaximumMemoryUse As Long  
    Get  
    Set
```

C++

```
public:  
static property long long MaximumMemoryUse {  
    long long get ();  
    void set (long long value);  
}
```

F#

```
static member MaximumMemoryUse : int64 with get, set
```

Property Value

Type: [Int64](#)

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.MemoryUsedByThisProcess Property

The amount of virtual memory, in bytes, allocated for this process.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static long MemoryUsedByThisProcess { get; }
```

VB

```
Public Shared ReadOnly Property MemoryUsedByThisProcess As Long  
    Get
```

C++

```
public:  
static property long long MemoryUsedByThisProcess {  
    long long get ();  
}
```

F#

```
static member MemoryUsedByThisProcess : int64 with get
```

Property Value

Type: [Int64](#)

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.MinimumAvailableMegaBytes Property

Decide the minimum amount of available memory before reducing page caching and flushing updated pages. Only used by 32-bit process.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong MinimumAvailableMegaBytes { get; set; }
```

VB

```
Public Property MinimumAvailableMegaBytes As ULong  
    Get  
    Set
```

C++

```
public:  
property unsigned long long MinimumAvailableMegaBytes {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
member MinimumAvailableMegaBytes : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.PageCacheEnabled Property

Enable or disable page cache. In cases where pages are not going to be revisited, performance may be improved by disabling the page cache. In other cases leave the page cache enabled (the default)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool PageCacheEnabled { get; set; }
```

VB

```
Public Property PageCacheEnabled As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool PageCacheEnabled {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member PageCacheEnabled : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also


[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.DataCache Methods

The [DataCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-----------------------------|
|  | Clear | Clear cache of cached pages |

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.Clear Method

Clear cache of cached pages

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Clear ()
```

VB

```
Public Sub Clear
```

C++

```
public:  
void Clear ()
```

F#

```
member Clear : unit -> unit
```

See Also


[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.DataCache Fields

The [DataCache](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | s_computerInfo | |

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

DataCache.s_computerInfo Field

[Missing <summary> documentation for "F:VelocityDb.DataCache.s_computerInfo"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ComputerInfo s_computerInfo
```

VB

```
Public Shared s_computerInfo As ComputerInfo
```

C++

```
public:  
static ComputerInfo^ s_computerInfo
```

F#

```
static val mutable s_computerInfo: ComputerInfo
```

Field Value

Type: [ComputerInfo](#)

See Also

[DataCache Class](#)

[VelocityDb Namespace](#)

IOptimizedPersistable Interface

add optimized persistence by implementing this interface. We provide [OptimizedPersistable](#) as a base class that implements this interface.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IOptimizedPersistable : IComparable
```

VB

```
Public Interface IOptimizedPersistable
    Inherits IComparable
```

C++




```
public interface class IOptimizedPersistable : IComparable
```










F#

```
type IOptimizedPersistable =
    interface
        interface IComparable
    end
```





The **IOptimizedPersistable** type exposes the following members.













Properties

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | By default true but override as returning false so that this type of objects isn't sharing pages with other type of objects. |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. |
|  FieldsLoaded | Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an override of LazyLoadFields returns true then the object members are loaded by calling [!:OptimizedPersistable.LoadFields] |

| | |
|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  FlushIfPageFull | By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page) |
|  Id | Gets the id of this object. The id is structured as Oid |
|  IsPersistent | Gets the persistent state of an object. An object is considered persistent when it has an Oid , that is the Id is not 0 |
|  IsUpdated | Gets the updated state of the object |
|  LazyLoadFields | By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading). |
|  MaxNumberOfDatabases | Possibly restrict instances of to a single Database. By default this property is UInt32.MaxValue but classes like BTreeSetOidShort, BTreeMapShortOid ... override this property to return 1 since short references are restricted to a single Database. |
|  ObjectsPerPage | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  PagesPerDatabase | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisting using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  PlacementDatabaseNumber | Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given. |
|  RemovedFromIndices | <code>true</code> if call to [!:Update(bool)] or UpdateObject(IOptimizedPersistable, Boolean, Boolean) caused this object to be removed from possible indices |

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  FlushTransients | Flushes batched up objects such as objects added with AddFast(Key) |
|  GetPage | Get the persistent storage Page of this object |
|  GetTypeVersion | The database engine needs this internally |
|  GetWrappedObject | Internally used when IOptimizedPersistable is a wrapper for a non IOptimizedPersistable |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  InitializeAfterRead | This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data. |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. |
|  Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
|  Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. |
|  PersistMyReferences | Persists references from this object |
|  ReadMe | Provides a way to customize how an object is read. Used by a code generator |
|  SetPage | Sets the persistent storage Page of this object |
|  SetTypeVersion | The database engine needs this internally |
|  ShallowCopyTo | Copies current object to a page |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. |
|  Update | Call this function before updating any fields of this object |
|  WriteMe | Provides a way to customize how an object is written. Used by a code generator |












See Also



[VelocityDb Namespace](#)

IOptimizedPersistable.IOptimizedPersistable Properties

The [IOptimizedPersistable](#) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | By default true but override as returning false so that this type of objects isn't sharing pages with other type of objects. |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. |
|  FieldsLoaded | Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an overrode of LazyLoadFields returns true then the object members are loaded by calling <code>[!:OptimizedPersistable.LoadFields]</code> |
|  FlushIfPageFull | By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page) |
|  Id | Gets the id of this object. The id is structured as Oid |
|  IsPersistent | Gets the persistent state of an object. An object is considered persistent when it has an Oid , that is the Id is not 0 |
|  IsUpdated | Gets the updated state of the object |
|  LazyLoadFields | By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading). |
|  MaxNumberOfDatabases | Possibly restrict instances of to a single Database. By default this property is <code>UInt32.MaxValue</code> but classes like <code>BTreeSetOidShort</code> , <code>BTreeMapShortOid</code> ... override this property to return 1 since short references are restricted to a single Database. |
|  ObjectsPerPage | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  PagesPerDatabase | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisting using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by |

| | | |
|-----------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  | PlacementDatabaseNumber | Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given. |
|  | RemovedFromIndices | true if call to [!:Update(bool)] or UpdateObject(IOptimizedPersistable, Boolean, Boolean) caused this object to be removed from possible indices |

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.AllowOtherTypesOnSamePage Property

By default true but override as returning false so that this type of objects isn't sharing pages with other type of objects.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool AllowOtherTypesOnSamePage { get; }
```

VB

```
ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
property bool AllowOtherTypesOnSamePage {  
    bool get ();  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Cache Property

By default, the an object cache is determined by a [SessionBase](#) constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
CacheEnum Cache { get; }
```

VB

```
ReadOnly Property Cache As CacheEnum  
Get
```

C++

```
property CacheEnum Cache {  
    CacheEnum get ();  
}
```

F#

```
abstract Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.FieldsLoaded Property

Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an override of [LazyLoadFields](#) returns true then the object members are loaded by calling **[!:OptimizedPersistable.LoadFields]**

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool FieldsLoaded { get; set; }
```

VB

```
Property FieldsLoaded As Boolean  
    Get  
    Set
```

C++

```
property bool FieldsLoaded {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract FieldsLoaded : bool with get, set
```

Property Value

Type: [Boolean](#)

true if all members are loaded; otherwise false

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.FlushIfPageFull Property

By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool FlushIfPageFull { get; }
```

VB

```
ReadOnly Property FlushIfPageFull As Boolean  
    Get
```

C++

```
property bool FlushIfPageFull {  
    bool get ();  
}
```

F#

```
abstract FlushIfPageFull : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Id Property

Gets the id of this object. The id is structured as [Oid](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ulong Id { get; set; }
```

VB

```
Property Id As ULong  
    Get  
    Set
```

C++

```
property unsigned long long Id {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
abstract Id : uint64 with get, set
```

Property Value

Type: [UInt64](#)

The id of this object or 0 if object is not persistent

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.IsPersistent Property

Gets the persistent state of an object. An object is considered persistent when it has an [Oid](#), that is the [Id](#) is not 0

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool IsPersistent { get; }
```

VB

```
ReadOnly Property IsPersistent As Boolean  
    Get
```

C++

```
property bool IsPersistent {  
    bool get ();  
}
```

F#

```
abstract IsPersistent : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.IsUpdated Property

Gets the updated state of the object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool IsUpdated { get; set; }
```

VB

```
Property IsUpdated As Boolean  
    Get  
    Set
```

C++

```
property bool IsUpdated {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract IsUpdated : bool with get, set
```

Property Value

Type: [Boolean](#)

true if updated; otherwise false

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.LazyLoadFields Property

By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool LazyLoadFields { get; }
```

VB

```
ReadOnly Property LazyLoadFields As Boolean  
    Get
```

C++

```
property bool LazyLoadFields {  
    bool get ();  
}
```

F#

```
abstract LazyLoadFields : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.MaxNumberOfDatabases Property

Possibly restrict instances of to a single Database. By default this property is UInt32.MaxValue but classes like BTreeSetOidShort, BTreeMapShortOid ... override this property to return 1 since short references are restricted to a single Database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
uint MaxNumberOfDatabases { get; }
```

VB

```
ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
property unsigned int MaxNumberOfDatabases {  
    unsigned int get ();  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.ObjectsPerPage Property

A default for number of objects per database page used when persisting objects without an explicit [Placement](#) object or if persisted using [Persist\(SessionBase, IOptimizedPersistable, Boolean, Boolean\)](#). This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ushort ObjectsPerPage { get; }
```

VB

```
ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
property unsigned short ObjectsPerPage {  
    unsigned short get ();  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get
```

Return Value

Type: [UInt16](#)

The requested number of objects per page.

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.PagesPerDatabase Property

A default for number of objects per database page used when persisting objects without an explicit [Placement](#) object or if persisting using [Persist\(SessionBase, IOptimizedPersistable, Boolean, Boolean\)](#). This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ushort PagesPerDatabase { get; }
```

VB

```
ReadOnly Property PagesPerDatabase As UShort  
    Get
```

C++

```
property unsigned short PagesPerDatabase {  
    unsigned short get ();  
}
```

F#

```
abstract PagesPerDatabase : uint16 with get
```

Return Value

Type: [UInt16](#)

The requested number of pages per database

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.PlacementDatabaseNumber Property

Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
uint PlacementDatabaseNumber { get; }
```

VB

```
ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
property unsigned int PlacementDatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.RemovedFromIndices Property

true

if call to **[!:Update(bool)]** or [UpdateObject\(IOptimizedPersistable, Boolean, Boolean\)](#) caused this object to be removed from possible indices

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool RemovedFromIndices { get; set; }
```

VB

```
Property RemovedFromIndices As Boolean  
    Get  
    Set
```

C++

```
property bool RemovedFromIndices {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract RemovedFromIndices : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.IOptimizedPersistable Methods

The [IOptimizedPersistable](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FlushTransients | Flushes batched up objects such as objects added with AddFast(Key) |
| GetPage | Get the persistent storage Page of this object |
| GetTypeVersion | The database engine needs this internally |
| GetWrappedObject | Internally used when IOptimizedPersistable is a wrapper for a non IOptimizedPersistable |
| InitializeAfterRead | This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data. |
| InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. |
| Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
| Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. |
| PersistMyReferences | Persists references from this object |
| ReadMe | Provides a way to customize how an object is read. Used by a code generator |
| SetPage | Sets the persistent storage Page of this object |
| SetTypeVersion | The database engine needs this internally |
| ShallowCopyTo | Copies current object to a page |
| Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. |
| Update | Call this function before updating any fields of this object |
| WriteMe | Provides a way to customize how an object is written. Used by a code generator |

VelocityDB Class Library

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.FlushTransients Method

Flushes batched up objects such as objects added with [AddFast\(Key\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void FlushTransients ()
```

VB

```
Sub FlushTransients
```

C++

```
void FlushTransients ()
```

F#

```
abstract FlushTransients : unit -> unit
```

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.GetPage Method

Get the persistent storage [Page](#) of this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Page GetPage ()
```

VB

```
Function GetPage As Page
```

C++

```
Page^ GetPage ()
```

F#

```
abstract GetPage : unit -> Page
```

Return Value

Type: [Page](#)

The persistent storage [Page](#) of this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.GetTypeVersion Method

The database engine needs this internally

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
TypeVersion GetTypeVersion()
```

VB

```
Function GetTypeVersion As TypeVersion
```

C++

```
TypeVersion^ GetTypeVersion()
```

F#

```
abstract GetTypeVersion : unit -> TypeVersion
```

Return Value

Type: [TypeVersion](#)

The [TypeVersion](#) of this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.GetWrappedObject Method

Internally used when [IOptimizedPersistable](#) is a wrapper for a non [IOptimizedPersistable](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object GetWrappedObject ()
```

VB

```
Function GetWrappedObject As Object
```

C++

```
Object^ GetWrappedObject ()
```

F#

```
abstract GetWrappedObject : unit -> Object
```

Return Value

Type: [Object](#)

Wrapped object if set; otherwise this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.InitializeAfterRead Method

This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
void InitializeAfterRead(  
    SessionBase^ session  
)
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void InitializeAfterRecreate (  
    SessionBase session  
)
```

VB

```
Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
void InitializeAfterRecreate (  
    SessionBase^ session  
)
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object



See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Persist Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-----------------------|
|  | Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
|  | Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. |

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Persist Method (SessionBase, IOptimizedPersistable, Boolean, Boolean)

Persists this object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ulong Persist(  
    SessionBase session,  
    IOptimizedPersistable placeHint,  
    bool persistRefs = true,  
    bool disableFlush = false  
)
```

VB

```
Function Persist (  
    session As SessionBase,  
    placeHint As IOptimizedPersistable,  
    Optional persistRefs As Boolean = true,  
    Optional disableFlush As Boolean = false  
) As ULong
```

C++

```
unsigned long long Persist(  
    SessionBase^ session,  
    IOptimizedPersistable^ placeHint,  
    bool persistRefs = true,  
    bool disableFlush = false  
)
```

F#

```
abstract Persist :  
    session : SessionBase *  
    placeHint : IOptimizedPersistable *  
    ?persistRefs : bool *  
    ?disableFlush : bool  
(* Defaults:  
    let _persistRefs = defaultArg persistRefs true  
    let _disableFlush = defaultArg disableFlush false  
*)  
-> uint64
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

placeHint

Type: [VelocityDb.IOptimizedPersistable](#)

Use placement as specified by this object type, see [PlacementDatabaseNumber](#), [ObjectsPerPage](#) and [PagesPerDatabase](#)

persistRefs (Optional)

Type: [System.Boolean](#)

Persist any referenced object now or delay until flush/commit

disableFlush (Optional)

Type: [System.Boolean](#)

Controls possible flushing of updated pages. Set to true if you want to prevent updated pages from being flushed to disk and setting such pages to a non updated state.

Return Value

Type: [UInt64](#)

The object id of the persistent object

See Also

[IOptimizedPersistable Interface](#)

[Persist Overload](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Persist Method (Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable))

Persists this object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
)
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of the persistent object

See Also

[IOptimizedPersistable Interface](#)

[Persist Overload](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.PersistMyReferences Method

Persists references from this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void PersistMyReferences (
    SessionBase session,
    bool inFlush
)
```

VB

```
Sub PersistMyReferences (
    session As SessionBase,
    inFlush As Boolean
)
```

C++

```
void PersistMyReferences (
    SessionBase^ session,
    bool inFlush
)
```

F#

```
abstract PersistMyReferences :
    session : SessionBase *
    inFlush : bool -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

inFlush

Type: [System.Boolean](#)

are we in a page flush

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.ReadMe Method

Provides a way to customize how an object is read. Used by a code generator

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
    int graphDepth,
```

```
    int graphDepthToLoad,  
    bool primitivesOnly  
)
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

the type version of the object being read

memberBytes

Type: [System.Byte\[\]](#)

the raw bytes to read the object from

offset

Type: [System.Int32](#)

current offset into the raw object bytes

session

Type: [VelocityDb.Session.SessionBase](#)

the active session

page

Type: [VelocityDb.Page](#)

the page of the object

useOidShort

Type: [System.Boolean](#)

is object using short references

schema

Type: [VelocityDb.TypeInfo.Schema](#)

the active schema

openRefs

VelocityDB Class Library

Type: [System.Boolean](#)

open references objects

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

a list of field members to load

graphDepth

Type: [System.Int32](#)

current depth in object read

graphDepthToLoad

Type: [System.Int32](#)

requested max object depth to read

primitivesOnly

Type: [System.Boolean](#)

if true, only load primitive members (not referenced objects)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.SetPage Method

Sets the persistent storage [Page](#) of this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void SetPage (  
    Page page  
)
```

VB

```
Sub SetPage (  
    page As Page  
)
```

C++

```
void SetPage (  
    Page^ page  
)
```

F#

```
abstract SetPage :  
    page : Page -> unit
```

Parameters

page

Type: [VelocityDb.Page](#)

A persistent storage [Page](#)

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.SetTypeVersion Method

The database engine needs this internally

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void SetTypeVersion(  
    TypeVersion shape  
)
```

VB

```
Sub SetTypeVersion (  
    shape As TypeVersion  
)
```

C++

```
void SetTypeVersion(  
    TypeVersion^ shape  
)
```

F#

```
abstract SetTypeVersion :  
    shape : TypeVersion -> unit
```

Parameters

shape

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The [TypeVersion](#) to use for this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.ShallowCopyTo Method

Copies current object to a page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
IOptimizedPersistable ShallowCopyTo(  
    Page page  
)
```

```
VB  
Function ShallowCopyTo (  
    page As Page  
) As IOptimizedPersistable
```

```
C++  
IOptimizedPersistable^ ShallowCopyTo(  
    Page^ page  
)
```

```
F#  
abstract ShallowCopyTo :  
    page : Page -> IOptimizedPersistable
```

Parameters

page

Type: [VelocityDb.Page](#)

The page to copy to

Return Value

Type: [IOptimizedPersistable](#)

Id of object copy

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Unpersist (  
    SessionBase session  
)
```

VB

```
Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
void Unpersist (  
    SessionBase^ session  
)
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.Update Method

Call this function before updating any fields of this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool Update ()
```

VB

```
Function Update As Boolean
```

C++

```
bool Update ()
```

F#

```
abstract Update : unit -> bool
```

Return Value

Type: [Boolean](#)

true if update was successful; otherwise false

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IOptimizedPersistable.WriteMe Method

Provides a way to customize how an object is written. Used by a code generator

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
byte[] WriteMe(  
    TypeVersion typeVersion,  
    bool addShapeNumber,  
    PageInfo pageInfo,  
    IOptimizedPersistable owner,  
    SessionBase session,  
    bool inFlush  
)
```

VB

```
Function WriteMe (  
    typeVersion As TypeVersion,  
    addShapeNumber As Boolean,  
    pageInfo As PageInfo,  
    owner As IOptimizedPersistable,  
    session As SessionBase,  
    inFlush As Boolean  
) As Byte ()
```

C++

```
array<unsigned char>^ WriteMe(  
    TypeVersion^ typeVersion,  
    bool addShapeNumber,  
    PageInfo^ pageInfo,  
    IOptimizedPersistable^ owner,  
    SessionBase^ session,  
    bool inFlush  
)
```

F#

```
abstract WriteMe :  
    typeVersion : TypeVersion *  
    addShapeNumber : bool *  
    pageInfo : PageInfo *  
    owner : IOptimizedPersistable *  
    session : SessionBase *  
    inFlush : bool -> byte[]
```

Parameters

typeVersion

VelocityDB Class Library

Type: [VelocityDb.TypeInfo.TypeVersion](#)

the type version of the object being written

addShapeNumber

Type: [System.Boolean](#)

add type version number to object bytes

pageInfo

Type: [VelocityDb.PageInfo](#)

the page info for the page to written to

owner

Type: [VelocityDb.IOptimizedPersistable](#)

owning object

session

Type: [VelocityDb.Session.SessionBase](#)

the active session

inFlush

Type: [System.Boolean](#)

are we in a page flush

Return Value

Type: [Byte\[\]](#)

Bytes representing this object

See Also

[IOptimizedPersistable Interface](#)

[VelocityDb Namespace](#)

IReferenceTracked Interface

Maintains a set of all [References](#) to this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IReferenceTracked : IOptimizedPersistable,  
    IComparable
```

VB

```
Public Interface IReferenceTracked  
    Inherits IOptimizedPersistable, IComparable
```

C++


```
public interface class IReferenceTracked : IOptimizedPersistable,  
    IComparable
```

F#

```
type IReferenceTracked =  
    interface  
        interface IOptimizedPersistable  
        interface IComparable  
    end
```

The **IReferenceTracked** type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------|
|  | References | Get a collection of references to this object |


See Also

[VelocityDb Namespace](#)

IReferenceTracked.IReferenceTracked Properties

The [IReferenceTracked](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------|
|  | References | Get a collection of references to this object |

See Also

[IReferenceTracked Interface](#)

[VelocityDb Namespace](#)

IReferenceTracked.References Property

Get a collection of references to this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
BTreeSet<Reference> References { get; }
```

VB

```
ReadOnly Property References As BTreeSet (Of Reference)  
Get
```

C++

```
property BTreeSet<Reference^>^ References {  
    BTreeSet<Reference^>^ get ();  
}
```

F#

```
abstract References : BTreeSet<Reference> with get
```

Property Value

Type: [BTreeSet\(Reference\)](#)

See Also

[IReferenceTracked Interface](#)

[VelocityDb Namespace](#)

License Class

Contains a list of possible license attributes for licensing VelocityDb class library and server

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.License

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class License : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class License
    Inherits OptimizedPersistable
```

C++







```
[SerializableAttribute]
public ref class License : public OptimizedPersistable
```






F#

```
[<SerializableAttribute>]
type License =
    class
        inherit OptimizedPersistable
    end
```



The **License** type exposes the following members.

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|---------------------------------------------------------------------|
|  | DateTimeCreated | Gets the time of license creation. |
|  | DomainName | Gets the restricted domain name |
|  | ExpireDate | Gets expire time. |
|  | HostName | Gets the restricted host name |
|  | MajorVersion | Gets/set the major version attribute of a license |
|  | MaxNumberOfClients | Gets the maximum number of Server clients permitted by this license |

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|  MinorVersion | Gets/set the minor version attribute of a license |
|  ObjectsPerPage | Store one License per Page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  PlacementDatabaseNumber | Gets the preferred Database number for licenses (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  ProcessorCount | Gets/sets maximum number of processors for which license is valid. |
|  UserName | Gets the user name restriction |



Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  WriteMe | (Overrides OptimizedPersistable.WriteMe(TypeVersion, Boolean, PageInfo, IOptimizedPersistable, SessionBase, Boolean).) |

Fields

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  LicenseCheckDatabase | The Database number of the database containing a VelocityDb license(s) |
|  PlaceInDatabase | Hint about a Database location for this type of object. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |












See Also

[VelocityDb Namespace](#)

License.License Properties

The [License](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|  | DateTimeCreated | Gets the time of license creation. |
|  | DomainName | Gets the restricted domain name |
|  | ExpireDate | Gets expire time. |
|  | HostName | Gets the restricted host name |
|  | MajorVersion | Gets/set the major version attribute of a license |
|  | MaxNumberOfClients | Gets the maximum number of Server clients permitted by this license |
|  | MinorVersion | Gets/set the minor version attribute of a license |
|  | ObjectsPerPage | Store one License per Page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | PlacementDatabaseNumber | Gets the preferred Database number for licenses (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  | ProcessorCount | Gets/sets maximum number of processors for which license is valid. |
|  | UserName | Gets the user name restriction |

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.DateTimeCreated Property

Gets the time of license creation.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DateTime DateTimeCreated { get; }
```

VB

```
Public ReadOnly Property DateTimeCreated As DateTime  
    Get
```

C++

```
public:  
property DateTime DateTimeCreated {  
    DateTime get ();  
}
```

F#

```
member DateTimeCreated : DateTime with get
```

Property Value

Type: [DateTime](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.DomainName Property

Gets the restricted domain name

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string DomainName { get; set; }
```

VB

```
Public Property DomainName As String  
    Get  
    Set
```

C++

```
public:  
property String^ DomainName {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member DomainName : string with get, set
```

Property Value

Type: [String](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.ExpireDate Property

Gets expire time.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DateTime ExpireDate { get; set; }
```

VB

```
Public Property ExpireDate As DateTime  
    Get  
    Set
```

C++

```
public:  
property DateTime ExpireDate {  
    DateTime get ();  
    void set (DateTime value);  
}
```

F#

```
member ExpireDate : DateTime with get, set
```

Property Value

Type: [DateTime](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.HostName Property

Gets the restricted host name

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string HostName { get; set; }
```

VB

```
Public Property HostName As String  
    Get  
    Set
```

C++

```
public:  
property String^ HostName {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member HostName : string with get, set
```

Property Value

Type: [String](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.MajorVersion Property

Gets/set the major version attribute of a license

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int MajorVersion { get; set; }
```

VB

```
Public Property MajorVersion As Integer  
    Get  
    Set
```

C++

```
public:  
property int MajorVersion {  
    int get ();  
    void set (int value);  
}
```

F#

```
member MajorVersion : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.MaxNumberOfClients Property

Gets the maximum number of Server clients permitted by this license

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint MaxNumberOfClients { get; set; }
```

VB

```
Public Property MaxNumberOfClients As UInteger  
    Get  
    Set
```

C++

```
public:  
property unsigned int MaxNumberOfClients {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
member MaxNumberOfClients : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.MinorVersion Property

Gets/set the minor version attribute of a license

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int MinorVersion { get; set; }
```

VB

```
Public Property MinorVersion As Integer  
    Get  
    Set
```

C++

```
public:  
property int MinorVersion {  
    int get ();  
    void set (int value);  
}
```

F#

```
member MinorVersion : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.ObjectsPerPage Property

Store one License per Page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Return Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.PlacementDatabaseNumber Property

Gets the preferred Database number for licenses

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint PlacementDatabaseNumber { get; }
```

VB

```
Public Overrides ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int PlacementDatabaseNumber {  
    unsigned int get () override;  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get  
override PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.PlacementDatabaseNumber](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.ProcessorCount Property

Gets/sets maximum number of processors for which license is valid.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int ProcessorCount { get; set; }
```

VB

```
Public Property ProcessorCount As Integer  
    Get  
    Set
```

C++

```
public:  
property int ProcessorCount {  
    int get ();  
    void set (int value);  
}
```

F#

```
member ProcessorCount : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.UserName Property

Gets the user name restriction

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string UserName { get; set; }
```

VB

```
Public Property UserName As String  
    Get  
    Set
```

C++

```
public:  
property String^ UserName {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member UserName : string with get, set
```

Property Value

Type: [String](#)

See Also



[License Class](#)

[VelocityDb Namespace](#)



License.License Methods

The [License](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  | WriteMe | (Overrides OptimizedPersistable.WriteMe(TypeVersion, Boolean, PageInfo, IOptimizedPersistable, SessionBase, Boolean).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
Page^ page,  
bool useOidShort,  
Schema^ schema,  
bool openRefs,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,Velo

`cityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]`

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.License.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.WriteMe Method

[Missing <summary> documentation for

"M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override byte[] WriteMe(  
    TypeVersion typeVersion,  
    bool addShapeNumber,  
    PageInfo pageInfo,  
    IOptimizedPersistable owner,  
    SessionBase session,  
    bool inFlush  
)
```

VB

```
Public Overrides Function WriteMe (  
    typeVersion As TypeVersion,  
    addShapeNumber As Boolean,  
    pageInfo As PageInfo,  
    owner As IOptimizedPersistable,  
    session As SessionBase,  
    inFlush As Boolean  
) As Byte ()
```

C++

```
public:  
virtual array<unsigned char>^ WriteMe(  
    TypeVersion^ typeVersion,  
    bool addShapeNumber,  
    PageInfo^ pageInfo,  
    IOptimizedPersistable^ owner,  
    SessionBase^ session,  
    bool inFlush  
) override
```

F#

```
abstract WriteMe :  
    typeVersion : TypeVersion *  
    addShapeNumber : bool *  
    pageInfo : PageInfo *  
    owner : IOptimizedPersistable *  
    session : SessionBase *  
    inFlush : bool -> byte[]  
override WriteMe :  
    typeVersion : TypeVersion *
```

```
addShapeNumber : bool *
pageInfo : PageInfo *
owner : IOptimizedPersistable *
session : SessionBase *
inFlush : bool -> byte[]
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

addShapeNumber

Type: [System.Boolean](#)

[Missing <param name="addShapeNumber"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

pageInfo

Type: [VelocityDb.PageInfo](#)

[Missing <param name="pageInfo"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

owner

Type: [VelocityDb.IOptimizedPersistable](#)

[Missing <param name="owner"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

inFlush

Type: [System.Boolean](#)

[Missing <param name="inFlush"/> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

Return Value

Type: [Byte\[\]](#)

[Missing <returns> documentation for "M:VelocityDb.License.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

VelocityDB Class Library

Implements

[IOptimizedPersistable.WriteMe\(TypeVersion, Boolean, PageInfo, IOptimizedPersistable, SessionBase, Boolean\)](#)

See Also



[License Class](#)

[VelocityDb Namespace](#)

License.License Fields

The [License](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|------------------------------------------------------------------------|
|  | LicenseCheckDatabase | The Database number of the database containing a VelocityDb license(s) |
|  | PlaceInDatabase | Hint about a Database location for this type of object. |

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.CheckDatabase Field

The Database number of the database containing a VelocityDb license(s)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint LicenseCheckDatabase = 4
```

VB

```
Public Const LicenseCheckDatabase As UInteger = 4
```

C++

```
public:  
literal unsigned int LicenseCheckDatabase = 4
```

F#

```
static val mutable LicenseCheckDatabase: uint32
```

Field Value

Type: [UInt32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

License.PlaceInDatabase Field

Hint about a Database location for this type of object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint PlaceInDatabase = 20
```

VB

```
Public Const PlaceInDatabase As UInteger = 20
```

C++

```
public:  
literal unsigned int PlaceInDatabase = 20
```

F#

```
static val mutable PlaceInDatabase: uint32
```

Field Value

Type: [UInt32](#)

See Also

[License Class](#)

[VelocityDb Namespace](#)

OfType Class

An object of this type is a base class of type returned by [AllObjects\(T\)\(Boolean, Boolean\)](#) or [AllObjects\(T\)\(Boolean\)](#)

Inheritance Hierarchy

[System.Object](#)

VelocityDb.OfType

[VelocityDb.AllObjects\(T\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class OfType : IEnumerable
```

VB

```
Public Class OfType
    Implements IEnumerable
```

C++



```
public ref class OfType : IEnumerable
```

F#



```
type OfType =
    class
        interface IEnumerable
    end
```

The **OfType** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------|
|  | OfType(Type, Database, Boolean) | Constructs this enumeration wrapper object |
|  | OfType(Type, SessionBase, Boolean, Boolean) | Constructs this enumeration wrapper object |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | Count | Get a count of instances of the given type. This value is computed by a quick iteration over all objects of the requested type. |
|  | GetEnumerator | Enumeration of all objects of the selected Type |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | ElementAt | Override LINQ for faster access (Defined by OfTypeExtension.) |
|  | Skip | Bypasses a specified number of elements in a sequence and then returns the remaining elements. (Defined by OfTypeExtension.) |

See Also

[VelocityDb Namespace](#)

OfType Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------|
|  | OfType(Type, Database, Boolean) | Constructs this enumeration wrapper object |
|  | OfType(Type, SessionBase, Boolean, Boolean) | Constructs this enumeration wrapper object |

See Also

[OfType Class](#)

[VelocityDb Namespace](#)

OfType Constructor (Type, Database, Boolean)

Constructs this enumeration wrapper object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OfType(  
    Type type,  
    Database db,  
    bool includeSubClasses = false  
)
```

VB

```
Public Sub New (  
    type As Type,  
    db As Database,  
    Optional includeSubClasses As Boolean = false  
)
```

C++

```
public:  
OfType(  
    Type^ type,  
    Database^ db,  
    bool includeSubClasses = false  
)
```

F#

```
new :  
    type : Type *  
    db : Database *  
    ?includeSubClasses : bool  
(* Defaults:  
    let_includeSubClasses = defaultArg includeSubClasses false  
)  
-> OfType
```

Parameters

type

Type: [System.Type](#)

The [Type](#) to look for.

db

Type: [VelocityDb.Database](#)

The active db

VelocityDB Class Library

includeSubClasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes

See Also

[OfType Class](#)

[OfType Overload](#)

[VelocityDb Namespace](#)

OfType Constructor (Type, SessionBase, Boolean, Boolean)

Constructs this enumeration wrapper object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OfType(  
    Type type,  
    SessionBase session,  
    bool includeSubclasses,  
    bool databasePerType  
)
```

VB

```
Public Sub New (  
    type As Type,  
    session As SessionBase,  
    includeSubclasses As Boolean,  
    databasePerType As Boolean  
)
```

C++

```
public:  
OfType(  
    Type^ type,  
    SessionBase^ session,  
    bool includeSubclasses,  
    bool databasePerType  
)
```

F#

```
new :  
    type : Type *  
    session : SessionBase *  
    includeSubclasses : bool *  
    databasePerType : bool -> OfType
```

Parameters

type

Type: [System.Type](#)

The [Type](#) of persisted objects to enumerate

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

VelocityDB Class Library

includeSubclasses

Type: [System.Boolean](#)

Also return instances of sub classes

databasePerType

Type: [System.Boolean](#)

Assume that persisted objects where made persistent the simple way using

[Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#). When persisting this way, each object type gets its own [Database](#) which makes finding these objects easier and faster

See Also

[OfType Class](#)



[OfType Overload](#)

[VelocityDb Namespace](#)



OfType.OfType Methods

The [OfType](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | Count | Get a count of instances of the given type. This value is computed by a quick iteration over all objects of the requested type. |
|  | GetEnumerator | Enumeration of all objects of the selected Type |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | ElementAt | Override LINQ for faster access (Defined by OfTypeExtension .) |
|  | Skip | Bypasses a specified number of elements in a sequence and then returns the remaining elements. (Defined by OfTypeExtension .) |

See Also

[OfType Class](#)

[VelocityDb Namespace](#)

OfType.Count Method

Get a count of instances of the given type. This value is computed by a quick iteration over all objects of the requested type.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ulong Count ()
```

VB

```
Public Overridable Function Count As ULong
```

C++

```
public:  
virtual unsigned long long Count ()
```

F#

```
abstract Count : unit -> uint64  
override Count : unit -> uint64
```

Return Value

Type: [UInt64](#)

Number of instances found of the given [Type](#)

See Also

[OfType Class](#)

[VelocityDb Namespace](#)

OfType.GetEnumerator Method

Enumeration of all objects of the selected [Type](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator
```

C++

```
public:  
virtual IEnumerator^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator  
override GetEnumerator : unit -> IEnumerator
```

Return Value

Type: [IEnumerator](#)

The enumeration

Implements

[IEnumerable.GetEnumerator\(\)](#)

See Also

[OfType Class](#)

[VelocityDb Namespace](#)

OfType.OfTpeIterator Class

Iterator for type [OfType](#)

Inheritance Hierarchy

[System.Object](#)

VelocityDb.OfTpe.OfTpeIterator

[VelocityDb.AllObjects\(T\).AllObjectsIterator](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class OfTpeIterator : IEnumerable
```

VB

```
Public Class OfTpeIterator  
    Implements IEnumerable
```

C++





```
public ref class OfTpeIterator : IEnumerable
```

F#

```
type OfTpeIterator =  
    class  
        interface IEnumerable  
    end
```

The OfTpe.OfTpeIterator type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------|
|  | GetEnumerator | Enumeration of all objects of type T |
|  | GetObjects | |
|  | SkipIterator | Bypasses a specified number of elements in a sequence and then returns the remaining elements. |
|  | TakeIterator | Returns a specified number of contiguous elements from the start of a sequence. |





See Also

[VelocityDb Namespace](#)

OfTypeIterator.OfTypeIterator Methods

The [OfType.OfTypeIterator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------|
|  | GetEnumerator | Enumeration of all objects of type T |
|  | GetObjects | |
|  | SkipIterator | Bypasses a specified number of elements in a sequence and then returns the remaining elements. |
|  | TakeIterator | Returns a specified number of contiguous elements from the start of a sequence. |

See Also

[OfType.OfTypeIterator Class](#)

[VelocityDb Namespace](#)

OfType.OfTypeIterator.GetEnumerator Method

Enumeration of all objects of type T

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator
```

C++

```
public:  
virtual IEnumerator^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator  
override GetEnumerator : unit -> IEnumerator
```

Return Value

Type: [IEnumerator](#)

The enumerator of T

Implements

[IEnumerable.GetEnumerator\(\)](#)

See Also

[OfType.OfTypeIterator Class](#)

[VelocityDb Namespace](#)

OfType.OfTypeIterator.GetObjects Method

[Missing <summary> documentation for "M:VelocityDb.OfType.OfTypeIterator.GetObjects"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable GetObjects ()
```

VB

```
Public Function GetObjects As IEnumerable
```

C++

```
public:  
IEnumerable^ GetObjects ()
```

F#

```
member GetObjects : unit -> IEnumerable
```

Return Value

Type: [IEnumerable](#)

[Missing <returns> documentation for "M:VelocityDb.OfType.OfTypeIterator.GetObjects"]

See Also

[OfType.OfTypeIterator Class](#)

[VelocityDb Namespace](#)

OfType.OfTypeIterator.SkipIterator Method

Bypasses a specified number of elements in a sequence and then returns the remaining elements.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public IEnumerable SkipIterator(  
    int count  
)
```

```
VB  
Public Function SkipIterator (  
    count As Integer  
) As IEnumerable
```

```
C++  
public:  
IEnumerable^ SkipIterator(  
    int count  
)
```

```
F#  
member SkipIterator :  
    count : int -> IEnumerable
```

Parameters

count

Type: [System.Int32](#)

The number of elements to skip before returning the remaining elements.

Return Value

Type: [IEnumerable](#)

An [IEnumerable](#) that contains the elements that occur after the specified index in the input sequence.

See Also

[OfType.OfTypeIterator Class](#)

[VelocityDb Namespace](#)

OfType.OfTypeIterator.TakeIterator Method

Returns a specified number of contiguous elements from the start of a sequence.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable TakeIterator(  
    int count  
)
```

VB

```
Public Function TakeIterator (  
    count As Integer  
) As IEnumerable
```

C++

```
public:  
IEnumerable^ TakeIterator(  
    int count  
)
```

F#

```
member TakeIterator :  
    count : int -> IEnumerable
```

Parameters

count

Type: [System.Int32](#)

The number of elements to return.

Return Value

Type: [IEnumerable](#)

An [IEnumerable](#) that contains the specified number of elements from the start of the input sequence.

See Also

[OfType.OfTypeIterator Class](#)

[VelocityDb Namespace](#)

OfTypeExtension Class

A few extensions to improve performance of Linq for Objects queries

Inheritance Hierarchy

[System.Object](#)

VelocityDb.OfTypeExtension

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class OfTypeExtension
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class OfTypeExtension
```

C++



```
[ExtensionAttribute]  
public ref class OfTypeExtension abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type OfTypeExtension = class end
```

The **OfTypeExtension** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------|
|  | ElementAt | Override LINQ for faster access |
|  | Skip | Bypasses a specified number of elements in a sequence and then returns the remaining elements. |





See Also

[VelocityDb Namespace](#)

OfTypeExtension.OfTypeExtension Methods

The [OfTypeExtension](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|------------------------------------------------------------------------------------------------|
|   | ElementAt | Override LINQ for faster access |
|   | Skip | Bypasses a specified number of elements in a sequence and then returns the remaining elements. |

See Also

[OfTypeExtension Class](#)

[VelocityDb Namespace](#)

OfTypeExtension.ElementAt Method

Override LINQ for faster access

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Object ElementAt(  
    this OfType source,  
    int index  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function ElementAt (  
    source As OfType,  
    index As Integer  
) As Object
```

C++

```
public:  
[ExtensionAttribute]  
static Object^ ElementAt(  
    OfType^ source,  
    int index  
)
```

F#

```
[<ExtensionAttribute>]  
static member ElementAt :  
    source : OfType *  
    index : int -> Object
```

Parameters

source

Type: [VelocityDb.OfType](#)

The source enumeration

index

Type: [System.Int32](#)

The index requested

Return Value

Type: [Object](#)

Element at requested index

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [OfType](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[OfTypeExtension Class](#)

[VelocityDb Namespace](#)

OfTypeExtension.Skip Method

Bypasses a specified number of elements in a sequence and then returns the remaining elements.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEnumerable Skip(  
    this OfType source,  
    int count  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function Skip (  
    source As OfType,  
    count As Integer  
) As IEnumerable
```

C++

```
public:  
[ExtensionAttribute]  
static IEnumerable^ Skip(  
    OfType^ source,  
    int count  
)
```

F#

```
[<ExtensionAttribute>]  
static member Skip :  
    source : OfType *  
    count : int -> IEnumerable
```

Parameters

source

Type: [VelocityDb.OfType](#)

The Type being extended

count

Type: [System.Int32](#)

The number of elements to skip before returning the remaining elements.

Return Value

Type: [IEnumerable](#)

An [IEnumerable](#) that contains the elements that occur after the specified index in the input sequence.

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [OfType](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[OfTypeExtension Class](#)

[VelocityDb Namespace](#)

Oid Structure

The object identifier containing a 32 bit database part, a 16 bit page part and a 16 bit page part. These parts are combined into a 64bit unsigned number [Id](#).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public struct Oid : IComparable<Oid>, IEqualityComparer<Oid>
```

VB

```
<SerializableAttribute>
Public Structure Oid
    Implements IComparable(Of Oid), IEqualityComparer(Of Oid)
```

C++




```
[SerializableAttribute]
public value class Oid : IComparable<Oid>,
    IEqualityComparer<Oid>
```

F#



```
[<SealedAttribute>]
[<SerializableAttribute>]
type Oid =
    struct
        interface IComparable<Oid>
        interface IEqualityComparer<Oid>
    end
```




The **Oid** type exposes the following members.

Constructors















| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------|
|  | Oid(UInt64) | Creates an Oid |
|  | Oid(UInt32, UInt32) | Creates an Oid given a Database number and an OidShort (page-slot) |
|  | Oid(UInt32, OidShort) | Creates an Oid given a Database number and an OidShort (page-slot) |

Properties





| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------|
|  | Database | Gets the database number part of the Oid |
|  | Id | Gets the underlying UInt64 that contains the encoded Oid. |

| | |
|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
|  IdShort | Gets the page-slot parts of the underlying UInt64 that contains the encoded Oid. |
|  Page | Gets the page number part of the Oid |
|  Slot | Gets the slot number part of the Oid |

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------|
|  AsString | | Builds a string with DatabaseNumber-PageNumber-SlotNumber |
|  CompareTo | | Compares two Oid objects by id |
|  DatabaseNumber | | Extracts the Database number |
|  Encode(UInt32, UInt32) | | Constructs a UInt64 from the components Database number and an encoded UInt32 contains page and slot number |
|  Encode(UInt32, UInt16, UInt16) | | Constructs a UInt64 from the components Database number, Page number and page number |
|  Equals | | Compares by Id |
|  GetHashCode() | | Computes a hash code based on Id. (Overrides ValueType.GetHashCode().) |
|  GetHashCode(Oid) | | Returns a hash code for the specified object. |
|  IdFromString | | |
|  PageNumber | | Extracts the Page number |
|  SamePageAs | | Find out if Page number is the same in two different UInt64 numbers (encoded Oid) |
|  SlotNumber(UInt64) | | Extracts the page number |
|  SlotNumber(UInt64, UInt16) | | Sets the page number |
|  ToString | | Builds a string with DatabaseNumber-PageNumber-SlotNumber (Overrides ValueType.ToString().) |

Operators

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------|------------------------------|
|  GreaterThan | | Compares the id of two Oid's |
|  GreaterThanOrEqual | | Compares the id of two Oid's |
|  LessThan | | Compares the id of two Oid's |
|  LessThanOrEqual | | Compares the id of two Oid's |




VelocityDB Class Library

See Also

[VelocityDb Namespace](#)

Oid Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------|
|  | Oid(UInt64) | Creates an Oid |
|  | Oid(UInt32, UInt32) | Creates an Oid given a Database number and an OidShort (page-slot) |
|  | Oid(UInt32, OidShort) | Creates an Oid given a Database number and an OidShort (page-slot) |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid Constructor (UInt64)

Creates an Oid

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid(  
    ulong id  
)
```

VB

```
Public Sub New (  
    id As ULong  
)
```

C++

```
public:  
Oid(  
    unsigned long long id  
)
```

F#

```
new :  
    id : uint64 -> Oid
```

Parameters

id

Type: [System.UInt64](#)

Highest 32 bit is Database number, then follows the 16 bit Page number and the 16 bit page number

See Also

[Oid Structure](#)

[Oid Overload](#)

[VelocityDb Namespace](#)

Oid Constructor (UInt32, UInt32)

Creates an Oid given a Database number and an OidShort (page-slot)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid(  
    uint dbId,  
    uint pageSlotId  
)
```

VB

```
Public Sub New (  
    dbId As UInteger,  
    pageSlotId As UInteger  
)
```

C++

```
public:  
Oid(  
    unsigned int dbId,  
    unsigned int pageSlotId  
)
```

F#

```
new :  
    dbId : uint32 *  
    pageSlotId : uint32 -> Oid
```

Parameters

dbId

Type: [System.UInt32](#)

The Database number

pageSlotId

Type: [System.UInt32](#)

The page and slot numbers as a UInt32

See Also

[Oid Structure](#)

[Oid Overload](#)

[VelocityDb Namespace](#)

Oid Constructor (UInt32, OidShort)

Creates an Oid given a Database number and an OidShort (page-slot)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid(  
    uint dbId,  
    OidShort pageSlotId  
)
```

VB

```
Public Sub New (  
    dbId As UInteger,  
    pageSlotId As OidShort  
)
```

C++

```
public:  
Oid(  
    unsigned int dbId,  
    OidShort pageSlotId  
)
```

F#

```
new :  
    dbId : uint32 *  
    pageSlotId : OidShort -> Oid
```

Parameters

dbId

Type: [System.UInt32](#)

The Database number

pageSlotId

Type: [VelocityDb.OidShort](#)

The page and slot numbers as an OidShort

See Also

[Oid Structure](#)






[Oid Overload](#)

[VelocityDb Namespace](#)

Oid.Oid Properties

The [Oid](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------|
|  | Database | Gets the database number part of the Oid |
|  | Id | Gets the underlying UInt64 that contains the encoded Oid. |
|  | IdShort | Gets the page-slot parts of the underlying UInt64 that contains the encoded Oid. |
|  | Page | Gets the page number part of the Oid |
|  | Slot | Gets the slot number part of the Oid |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Database Property

Gets the database number part of the Oid

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint Database { get; }
```

VB

```
Public ReadOnly Property Database As UInteger  
    Get
```

C++

```
public:  
property unsigned int Database {  
    unsigned int get ();  
}
```

F#

```
member Database : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Id Property

Gets the underlying UInt64 that contains the encoded Oid.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Id { get; }
```

VB

```
Public ReadOnly Property Id As ULong  
    Get
```

C++

```
public:  
property unsigned long long Id {  
    unsigned long long get ();  
}
```

F#

```
member Id : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.IdShort Property

Gets the page-slot parts of the underlying UInt64 that contains the encoded Oid.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint IdShort { get; }
```

VB

```
Public ReadOnly Property IdShort As UInteger  
    Get
```

C++

```
public:  
property unsigned int IdShort {  
    unsigned int get ();  
}
```

F#

```
member IdShort : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Page Property

Gets the page number part of the Oid

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort Page { get; }
```

VB

```
Public ReadOnly Property Page As UShort  
    Get
```

C++

```
public:  
property unsigned short Page {  
    unsigned short get ();  
}
```

F#

```
member Page : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Slot Property

Gets the slot number part of the Oid

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort Slot { get; }
```

VB

```
Public ReadOnly Property Slot As UShort  
    Get
```

C++

```
public:  
property unsigned short Slot {  
    unsigned short get ();  
}
```

F#

```
member Slot : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also














[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Oid Methods

The [Oid](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
|  | AsString | Builds a string with DatabaseNumber-PageNumber-SlotNumber |
|  | CompareTo | Compares two Oid objects by id |
|  | DatabaseNumber | Extracts the Database number |
|  | Encode(UInt32, UInt32) | Constructs a UInt64 from the components Database number and an encoded UInt32 contains page and slot number |
|  | Encode(UInt32, UInt16, UInt16) | Constructs a UInt64 from the components Database number, Page number and page number |
|  | Equals | Compares by Id |
|  | GetHashCode() | Computes a hash code based on Id. (Overrides ValueType.GetHashCode().) |
|  | GetHashCode(Oid) | Returns a hash code for the specified object. |
|  | IdFromString | |
|  | PageNumber | Extracts the Page number |
|  | SamePageAs | Find out if Page number is the same in two different UInt64 numbers (encoded Oid) |
|  | SlotNumber(UInt64) | Extracts the page number |
|  | SlotNumber(UInt64, UInt16) | Sets the page number |
| | ToString | Builds a string with DatabaseNumber-PageNumber-SlotNumber (Overrides ValueType.ToString().) |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.AsString Method

Builds a string with DatabaseNumber-PageNumber-SlotNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string AsString(  
    ulong id  
)
```

VB

```
Public Shared Function AsString (  
    id As ULong  
) As String
```

C++

```
public:  
static String^ AsString(  
    unsigned long long id  
)
```

F#

```
static member AsString :  
    id : uint64 -> string
```

Parameters

id

Type: [System.UInt64](#)

A combined Database number, Page number and page number

Return Value

Type: [String](#)

DatabaseNumber-PageNumber-SlotNumber

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.CompareTo Method

Compares two Oid objects by id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    Oid otherOid  
)
```

VB

```
Public Function CompareTo (  
    otherOid As Oid  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Oid otherOid  
) sealed
```

F#

```
abstract CompareTo :  
    otherOid : Oid -> int  
override CompareTo :  
    otherOid : Oid -> int
```

Parameters

otherOid

Type: [VelocityDb.Oid](#)

[Missing <param name="otherOid"/> documentation for "M:VelocityDb.Oid.CompareTo(VelocityDb.Oid)"]

Return Value

Type: [Int32](#)

a negative number if less, 0 if equal or else a positive number

Implements

[IComparable\(T\).CompareTo\(T\)](#)

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.DatabaseNumber Method

Extracts the Database number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static uint DatabaseNumber(  
    ulong id  
)
```

```
VB  
Public Shared Function DatabaseNumber (  
    id As ULong  
) As UInteger
```

```
C++  
public:  
static unsigned int DatabaseNumber(  
    unsigned long long id  
)
```

```
F#  
static member DatabaseNumber :  
    id : uint64 -> uint32
```

Parameters

id

Type: [System.UInt64](#)

The combined Database number, Page number and page number

Return Value

Type: [UInt32](#)

The Database number





See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Encode Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
|   | Encode(UInt32, UInt32) | Constructs a UInt64 from the components Database number and an encoded UInt32 contains page and slot number |
|   | Encode(UInt32, UInt16, UInt16) | Constructs a UInt64 from the components Database number, Page number and page number |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Encode Method (UInt32, UInt32)

Constructs a UInt64 from the components Database number and an encoded UInt32 contains page and slot number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ulong Encode(  
    uint db,  
    uint pageSlot  
)
```

VB

```
Public Shared Function Encode (  
    db As UInteger,  
    pageSlot As UInteger  
) As ULong
```

C++

```
public:  
static unsigned long long Encode(  
    unsigned int db,  
    unsigned int pageSlot  
)
```

F#

```
static member Encode :  
    db : uint32 *  
    pageSlot : uint32 -> uint64
```

Parameters

db

Type: [System.UInt32](#)

Database number

pageSlot

Type: [System.UInt32](#)

Encoded page and slot number

Return Value

Type: [UInt64](#)

A number containing the full object identifier

VelocityDB Class Library

See Also

[Oid Structure](#)

[Encode Overload](#)

[VelocityDb Namespace](#)

Oid.Encode Method (UInt32, UInt16, UInt16)

Constructs a UInt64 from the components Database number, Page number and page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ulong Encode(  
    uint db,  
    ushort page,  
    ushort slot  
)
```

VB

```
Public Shared Function Encode (  
    db As UInteger,  
    page As UShort,  
    slot As UShort  
) As ULong
```

C++

```
public:  
static unsigned long long Encode(  
    unsigned int db,  
    unsigned short page,  
    unsigned short slot  
)
```

F#

```
static member Encode :  
    db : uint32 *  
    page : uint16 *  
    slot : uint16 -> uint64
```

Parameters

db

Type: [System.UInt32](#)

A Database number

page

Type: [System.UInt16](#)

A Page number

slot

Type: [System.UInt16](#)

A slot number

VelocityDB Class Library

Return Value

Type: [UInt64](#)

**[Missing <returns> documentation for
"M:VelocityDb.Oid.Encode(System.UInt32,System.UInt16,System.UInt16)"]**

See Also

[Oid Structure](#)

[Encode Overload](#)

[VelocityDb Namespace](#)

Oid.Equals Method

Compares by Id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    Oid x,  
    Oid y  
)
```

VB

```
Public Function Equals (  
    x As Oid,  
    y As Oid  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Oid x,  
    Oid y  
) sealed
```

F#

```
abstract Equals :  
    x : Oid *  
    y : Oid -> bool  
override Equals :  
    x : Oid *  
    y : Oid -> bool
```

Parameters

x

Type: [VelocityDb.Oid](#)

First object

y

Type: [VelocityDb.Oid](#)

Second object

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.Oid.Equals(VelocityDb.Oid,VelocityDb.Oid)"]

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)



See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.GetHashCode Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------|
|  | GetHashCode() | Computes a hash code based on Id. (Overrides ValueType.GetHashCode().) |
|  | GetHashCode(Oid) | Returns a hash code for the specified object. |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.GetHashCode Method

Computes a hash code based on Id.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDb.Oid.GetHashCode"]

See Also

[Oid Structure](#)

[GetHashCode Overload](#)

[VelocityDb Namespace](#)

Oid.GetHashCode Method (Oid)

Returns a hash code for the specified object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode (  
    Oid a  
)
```

VB

```
Public Function GetHashCode (  
    a As Oid  
) As Integer
```

C++

```
public:  
virtual int GetHashCode (  
    Oid a  
) sealed
```

F#

```
abstract GetHashCode :  
    a : Oid -> int  
override GetHashCode :  
    a : Oid -> int
```

Parameters

a

Type: [VelocityDb.Oid](#)

[Missing <param name="a"/> documentation for "M:VelocityDb.Oid.GetHashCode(VelocityDb.Oid)"]

Return Value

Type: [Int32](#)

A hash code for the specified object.

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| ArgumentNullException | The type of <i>obj</i> is a reference type and <i>obj</i> is a null reference (Nothing in Visual Basic). |

See Also

[Oid Structure](#)

[GetHashCode Overload](#)

[VelocityDb Namespace](#)

Oid.IdFromString Method

[Missing <summary> documentation for "M:VelocityDb.Oid.IdFromString(System.String)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ulong IdFromString(  
    string oidAsString  
)
```

VB

```
Public Shared Function IdFromString (  
    oidAsString As String  
) As ULong
```

C++

```
public:  
static unsigned long long IdFromString(  
    String^ oidAsString  
)
```

F#

```
static member IdFromString :  
    oidAsString : string -> uint64
```

Parameters

oidAsString

Type: [System.String](#)

[Missing <param name="oidAsString"/> documentation for "M:VelocityDb.Oid.IdFromString(System.String)"]

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for "M:VelocityDb.Oid.IdFromString(System.String)"]

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.PageNumber Method

Extracts the Page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort PageNumber(  
    ulong id  
)
```

VB

```
Public Shared Function PageNumber (  
    id As ULong  
) As UShort
```

C++

```
public:  
static unsigned short PageNumber(  
    unsigned long long id  
)
```

F#

```
static member PageNumber :  
    id : uint64 -> uint16
```

Parameters

id

Type: [System.UInt64](#)

The combined Database number, Page number and page number

Return Value

Type: [UInt16](#)

The Page number

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.SamePageAs Method

Find out if Page number is the same in two different UInt64 numbers (encoded Oid)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool SamePageAs (  
    ulong id,  
    ulong anId  
)
```

VB

```
Public Shared Function SamePageAs (  
    id As ULong,  
    anId As ULong  
) As Boolean
```

C++

```
public:  
static bool SamePageAs (  
    unsigned long long id,  
    unsigned long long anId  
)
```

F#

```
static member SamePageAs :  
    id : uint64 *  
    anId : uint64 -> bool
```

Parameters

id

Type: [System.UInt64](#)

A combined Database number, Page number and page number

anId

Type: [System.UInt64](#)

Another combined Database number, Page number and skot number

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDb.Oid.SamePageAs(System.UInt64,System.UInt64)"]

VelocityDB Class Library

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.SlotNumber Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|--------------------------|
|  | SlotNumber(UInt64) | Extracts the page number |
|  | SlotNumber(UInt64, UInt16) | Sets the page number |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.SlotNumber Method (UInt64)

Extracts the page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort SlotNumber(  
    ulong id  
)
```

VB

```
Public Shared Function SlotNumber (  
    id As ULong  
) As UShort
```

C++

```
public:  
static unsigned short SlotNumber(  
    unsigned long long id  
)
```

F#

```
static member SlotNumber :  
    id : uint64 -> uint16
```

Parameters

id

Type: [System.UInt64](#)

A combined Database number, Page number and page number

Return Value

Type: [UInt16](#)

A page number

See Also

[Oid Structure](#)

[SlotNumber Overload](#)

[VelocityDb Namespace](#)

Oid.SlotNumber Method (UInt64, UInt16)

Sets the page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static ulong SlotNumber(  
    ulong oid,  
    ushort slotNumber  
)
```

```
VB  
Public Shared Function SlotNumber (  
    oid As ULong,  
    slotNumber As UShort  
) As ULong
```

```
C++  
public:  
static unsigned long long SlotNumber(  
    unsigned long long oid,  
    unsigned short slotNumber  
)
```

```
F#  
static member SlotNumber :  
    oid : uint64 *  
    slotNumber : uint16 -> uint64
```

Parameters

oid

Type: [System.UInt64](#)

A combined Database number, Page number and page number

slotNumber

Type: [System.UInt16](#)

A slot number to use in the combined UInt64 value

Return Value

Type: [UInt64](#)

A combined Database number, Page number and an updated page number

See Also

[Oid Structure](#)

VelocityDB Class Library

[SlotNumber Overload](#)

[VelocityDb Namespace](#)

Oid.ToString Method

Builds a string with DatabaseNumber-PageNumber-SlotNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

DatabaseNumber-PageNumber-SlotNumber

See Also





[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.Oid Operators

The [Oid](#) type exposes the following members.

Operators

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|------------------------------|
|  | Greater Than | Compares the id of two Oid's |
|  | Greater Than Or Equal | Compares the id of two Oid's |
|  | Less Than | Compares the id of two Oid's |
|  | Less Than Or Equal | Compares the id of two Oid's |

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.GreaterThan Operator

Compares the id of two Oid's

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool operator >(
    Oid a,
    Oid b
)
```

VB

```
Public Shared Operator > (
    a As Oid,
    b As Oid
) As Boolean
```

C++

```
public:
static bool operator >(
    Oid a,
    Oid b
)
```

F#

```
static let inline (>)
    a : Oid *
    b : Oid : bool
```

Parameters

a

Type: [VelocityDb.Oid](#)

First Oid

b

Type: [VelocityDb.Oid](#)

Second Oid

Return Value

Type: [Boolean](#)

a.id > b.id

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.GreaterThanOrEqual Operator

Compares the id of two Oid's

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static bool operator >=(  
    Oid a,  
    Oid b  
)
```

```
VB  
Public Shared Operator >= (  
    a As Oid,  
    b As Oid  
) As Boolean
```

```
C++  
public:  
static bool operator >=(  
    Oid a,  
    Oid b  
)
```

```
F#  
static let inline (>=)  
    a : Oid *  
    b : Oid : bool
```

Parameters

a

Type: [VelocityDb.Oid](#)

First Oid

b

Type: [VelocityDb.Oid](#)

Second Oid

Return Value

Type: [Boolean](#)

a.id >= b.id

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.LessThan Operator

Compares the id of two Oid's

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool operator <(
    Oid a,
    Oid b
)
```

VB

```
Public Shared Operator < (
    a As Oid,
    b As Oid
) As Boolean
```

C++

```
public:
static bool operator <(
    Oid a,
    Oid b
)
```

F#

```
static let inline (<)
    a : Oid *
    b : Oid : bool
```

Parameters

a

Type: [VelocityDb.Oid](#)

First Oid

b

Type: [VelocityDb.Oid](#)

Second Oid

Return Value

Type: [Boolean](#)

a.id less than b.id

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

Oid.LessThanOrEqual Operator

Compares the id of two Oid's

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static bool operator <=(  
    Oid a,  
    Oid b  
)
```

```
VB  
Public Shared Operator <= (  
    a As Oid,  
    b As Oid  
) As Boolean
```

```
C++  
public:  
static bool operator <=(  
    Oid a,  
    Oid b  
)
```

```
F#  
static let inline (<=)  
    a : Oid *  
    b : Oid : bool
```

Parameters

a

Type: [VelocityDb.Oid](#)

First Oid

b

Type: [VelocityDb.Oid](#)

Second Oid

Return Value

Type: [Boolean](#)

a.id less or equal to b.id

See Also

[Oid Structure](#)

[VelocityDb Namespace](#)

OidShort Structure

The short object identifier containing a 16 bit page part and a 16 bit page part. This is used for short references within a single databases.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public struct OidShort : IComparable<OidShort>,
    IEqualityComparer<OidShort>
```

VB

```
<SerializableAttribute>
Public Structure OidShort
    Implements IComparable(Of OidShort), IEqualityComparer(Of OidShort)
```

C++



```
[SerializableAttribute]
public value class OidShort : IComparable<OidShort>,
    IEqualityComparer<OidShort>
```

F#


```
[<SealedAttribute>]
[<SerializableAttribute>]
type OidShort =
    struct
        interface IComparable<OidShort>
        interface IEqualityComparer<OidShort>
    end
```

The **OidShort** type exposes the following members.














Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------|
|  | OidShort(UInt32) | Constructs an OidShort given a 32bit number |
|  | OidShort(UInt64) | Constructs an OidShort given a 64bit number by using the 32 lower bits |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------|---------------------------------------|
|  | Id | Gets the OidShort encoded as a UInt32 |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | AsString | Builds a string with DatabaseNumber-PageNumber-SlotNumber |
|  | Compare | Compares by Id |
|  | CompareTo | Compares two OidShort objects by id |
|  | Encode | Merges a page number and a page number into a single number |
|  | Equals(Object) | Indicates whether this instance and a specified object are equal. (Overrides ValueType.Equals(Object) .) |
|  | Equals(OidShort, OidShort) | Determines whether the specified objects are equal. |
|  | GetHashCode() | Returns the hash code for this instance. (Overrides ValueType.GetHashCode() .) |
|  | GetHashCode(OidShort) | Returns a hash code for the specified object. |
|  | PageNumber | Extracts a Page number from a combined Page-Slot number |
|  | SamePageAs | Determines if two encoded OidShort have the same Page number |
|  | SlotNumber(UInt32) | Extracts a page number |
|  | SlotNumber(UInt32, UInt16) | Sets a page number |
|  | ToString | Builds a string with PageNumber-SlotNumber (Overrides ValueType.ToString() .) |

Operators



| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-----------------------------|
|  | Equality | Compares id of two OidShort |
|  | Inequality | Compares id of two OidShort |

See Also

[VelocityDb Namespace](#)

OidShort Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------|
|  | OidShort(UInt32) | Constructs an OidShort given a 32bit number |
|  | OidShort(UInt64) | Constructs an OidShort given a 64bit number by using the 32 lower bits |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort Constructor (UInt32)

Constructs an OidShort given a 32bit number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OidShort(  
    uint id  
)
```

VB

```
Public Sub New (  
    id As UInteger  
)
```

C++

```
public:  
OidShort(  
    unsigned int id  
)
```

F#

```
new :  
    id : uint32 -> OidShort
```

Parameters

id

Type: [System.UInt32](#)

The page and slot encoded Id from which to construct the OidShort

See Also

[OidShort Structure](#)

[OidShort Overload](#)

[VelocityDb Namespace](#)

OidShort Constructor (UInt64)

Constructs an OidShort given a 64bit number by using the 32 lower bits

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OidShort (
    ulong id
)
```

VB

```
Public Sub New (
    id As ULong
)
```

C++

```
public:
OidShort (
    unsigned long long id
)
```

F#

```
new :
    id : uint64 -> OidShort
```

Parameters

id

Type: [System.UInt64](#)

The full encoded Id from which to construct the OidShort

See Also

[OidShort Structure](#)


[OidShort Overload](#)

[VelocityDb Namespace](#)

OidShort.OidShort Properties

The [OidShort](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|---------------------------------------|
|  | Id | Gets the OidShort encoded as a UInt32 |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Id Property

Gets the OidShort encoded as a UInt32

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint Id { get; }
```

VB

```
Public ReadOnly Property Id As UIntInteger  
    Get
```

C++

```
public:  
property unsigned int Id {  
    unsigned int get ();  
}
```

F#

```
member Id : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also














[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.OidShort Methods

The [OidShort](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | AsString | Builds a string with DatabaseNumber-PageNumber-SlotNumber |
|  | Compare | Compares by Id |
|  | CompareTo | Compares two OidShort objects by id |
|  | Encode | Merges a page number and a page number into a single number |
|  | Equals(Object) | Indicates whether this instance and a specified object are equal. (Overrides ValueType.Equals(Object) .) |
|  | Equals(OidShort, OidShort) | Determines whether the specified objects are equal. |
|  | GetHashCode() | Returns the hash code for this instance. (Overrides ValueType.GetHashCode() .) |
|  | GetHashCode(OidShort) | Returns a hash code for the specified object. |
|  | PageNumber | Extracts a Page number from a combined Page-Slot number |
|  | SamePageAs | Determines if two encoded OidShort have the same Page number |
|  | SlotNumber(UInt32) | Extracts a page number |
|  | SlotNumber(UInt32, UInt16) | Sets a page number |
|  | ToString | Builds a string with PageNumber-SlotNumber (Overrides ValueType.ToString() .) |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.AsString Method

Builds a string with DatabaseNumber-PageNumber-SlotNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string AsString(  
    uint id  
)
```

VB

```
Public Shared Function AsString (  
    id As UInteger  
) As String
```

C++

```
public:  
static String^ AsString(  
    unsigned int id  
)
```

F#

```
static member AsString :  
    id : uint32 -> string
```

Parameters

id

Type: [System.UInt32](#)

A combined Page number and page number

Return Value

Type: [String](#)

PageNumber-SlotNumber

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Compare Method

Compares by Id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int Compare(  
    OidShort aId,  
    OidShort bId  
)
```

VB

```
Public Shared Function Compare (  
    aId As OidShort,  
    bId As OidShort  
) As Integer
```

C++

```
public:  
static int Compare(  
    OidShort aId,  
    OidShort bId  
)
```

F#

```
static member Compare :  
    aId : OidShort *  
    bId : OidShort -> int
```

Parameters

aId

Type: [VelocityDb.OidShort](#)

First object

bId

Type: [VelocityDb.OidShort](#)

Second object

Return Value

Type: [Int32](#)

Comparison value, -1 for less, 0 for equal and 1 for greater than

See Also

[OidShort Structure](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

OidShort.CompareTo Method

Compares two OidShort objects by id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    OidShort obj  
)
```

VB

```
Public Function CompareTo (  
    obj As OidShort  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    OidShort obj  
) sealed
```

F#

```
abstract CompareTo :  
    obj : OidShort -> int  
override CompareTo :  
    obj : OidShort -> int
```

Parameters

obj

Type: [VelocityDb.OidShort](#)

The object to compare with

Return Value

Type: [Int32](#)

a negative number if less, 0 if equal or else a positive number

Implements

[IComparable\(T\).CompareTo\(T\)](#)

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Encode Method

Merges a page number and a page number into a single number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static uint Encode(  
    ushort page,  
    ushort slot  
)
```

VB

```
Public Shared Function Encode (  
    page As UShort,  
    slot As UShort  
) As UInteger
```

C++

```
public:  
static unsigned int Encode(  
    unsigned short page,  
    unsigned short slot  
)
```

F#

```
static member Encode :  
    page : uint16 *  
    slot : uint16 -> uint32
```

Parameters

page

Type: [System.UInt16](#)

A Page number

slot

Type: [System.UInt16](#)

A slot number

Return Value

Type: [UInt32](#)

The combined Page and page number

See Also



[OidShort Structure](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

OidShort.Equals Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | Equals(Object) | Indicates whether this instance and a specified object are equal. (Overrides ValueType.Equals(Object) .) |
|  | Equals(OidShort, OidShort) | Determines whether the specified objects are equal. |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Equals Method (Object)

Indicates whether this instance and a specified object are equal.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to compare with the current instance.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if *obj* and this instance are the same type and represent the same value; otherwise, `false` (`False` in Visual Basic).

See Also

[OidShort Structure](#)

[Equals Overload](#)

[VelocityDb Namespace](#)

OidShort.Equals Method (OidShort, OidShort)

Determines whether the specified objects are equal.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    OidShort x,  
    OidShort y  
)
```

VB

```
Public Function Equals (  
    x As OidShort,  
    y As OidShort  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    OidShort x,  
    OidShort y  
) sealed
```

F#

```
abstract Equals :  
    x : OidShort *  
    y : OidShort -> bool  
override Equals :  
    x : OidShort *  
    y : OidShort -> bool
```

Parameters

x

Type: [VelocityDb.OidShort](#)

The first object of type *T* to compare.

y

Type: [VelocityDb.OidShort](#)

The second object of type *T* to compare.

Return Value

Type: [Boolean](#)

`true` (True in Visual Basic) if the specified objects are equal; otherwise, `false` (False in Visual Basic).

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)

See Also



[OidShort Structure](#)

[Equals Overload](#)

[VelocityDb Namespace](#)

OidShort.GetHashCode Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------|
|  | GetHashCode() | Returns the hash code for this instance. (Overrides ValueType.GetHashCode().) |
|  | GetHashCode(OidShort) | Returns a hash code for the specified object. |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.GetHashCode Method

Returns the hash code for this instance.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

A 32-bit signed integer that is the hash code for this instance.

See Also

[OidShort Structure](#)

[GetHashCode Overload](#)

[VelocityDb Namespace](#)

OidShort.GetHashCode Method (OidShort)

Returns a hash code for the specified object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode(  
    OidShort a  
)
```

VB

```
Public Function GetHashCode (  
    a As OidShort  
) As Integer
```

C++

```
public:  
virtual int GetHashCode(  
    OidShort a  
) sealed
```

F#

```
abstract GetHashCode :  
    a : OidShort -> int  
override GetHashCode :  
    a : OidShort -> int
```

Parameters

a

Type: [VelocityDb.OidShort](#)

[Missing <param name="a"/> documentation for "M:VelocityDb.OidShort.GetHashCode(VelocityDb.OidShort)"]

Return Value

Type: [Int32](#)

A hash code for the specified object.

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|------------------------------------------------------------------------------------------------------------------|
| ArgumentNullException | The type of <i>obj</i> is a reference type and <i>obj</i> is a null reference (<i>Nothing</i> in Visual Basic). |

See Also

[OidShort Structure](#)

[GetHashCode Overload](#)

[VelocityDb Namespace](#)

OidShort.PageNumber Method

Extracts a Page number from a combined Page-Slot number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static ushort PageNumber(  
    uint id  
)
```

```
VB  
Public Shared Function PageNumber (  
    id As UInteger  
) As UShort
```

```
C++  
public:  
static unsigned short PageNumber(  
    unsigned int id  
)
```

```
F#  
static member PageNumber :  
    id : uint32 -> uint16
```

Parameters

id

Type: [System.UInt32](#)

Combined Page and page number

Return Value

Type: [UInt16](#)

A Page number

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.SamePageAs Method

Determines if two encoded OidShort have the same Page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool SamePageAs (  
    uint id,  
    uint anId  
)
```

VB

```
Public Shared Function SamePageAs (  
    id As UInteger,  
    anId As UInteger  
) As Boolean
```

C++

```
public:  
static bool SamePageAs (  
    unsigned int id,  
    unsigned int anId  
)
```

F#

```
static member SamePageAs :  
    id : uint32 *  
    anId : uint32 -> bool
```

Parameters

id

Type: [System.UInt32](#)

[Missing <param name="id"/> documentation for

"M:VelocityDb.OidShort.SamePageAs(System.UInt32,System.UInt32)"]

anId

Type: [System.UInt32](#)

[Missing <param name="anId"/> documentation for

"M:VelocityDb.OidShort.SamePageAs(System.UInt32,System.UInt32)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDb.OidShort.SamePageAs(System.UInt32,System.UInt32)"]

VelocityDB Class Library

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.SlotNumber Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|------------------------|
|  | SlotNumber(UInt32) | Extracts a page number |
|  | SlotNumber(UInt32, UInt16) | Sets a page number |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.SlotNumber Method (UInt32)

Extracts a page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort SlotNumber(  
    uint id  
)
```

VB

```
Public Shared Function SlotNumber (  
    id As UInteger  
) As UShort
```

C++

```
public:  
static unsigned short SlotNumber(  
    unsigned int id  
)
```

F#

```
static member SlotNumber :  
    id : uint32 -> uint16
```

Parameters

id

Type: [System.UInt32](#)

Combined Page and page number

Return Value

Type: [UInt16](#)

A page number

See Also

[OidShort Structure](#)

[SlotNumber Overload](#)

[VelocityDb Namespace](#)

OidShort.SlotNumber Method (UInt32, UInt16)

Sets a page number

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static ulong SlotNumber(  
    uint oid,  
    ushort slotNumber  
)
```

```
VB  
Public Shared Function SlotNumber (  
    oid As UInteger,  
    slotNumber As UShort  
) As ULong
```

```
C++  
public:  
static unsigned long long SlotNumber(  
    unsigned int oid,  
    unsigned short slotNumber  
)
```

```
F#  
static member SlotNumber :  
    oid : uint32 *  
    slotNumber : uint16 -> uint64
```

Parameters

oid

Type: [System.UInt32](#)

Combined Page and page number

slotNumber

Type: [System.UInt16](#)

Slot number to use

Return Value

Type: [UInt64](#)

A combined Page number with an updated page number

See Also

[OidShort Structure](#)

VelocityDB Class Library

[SlotNumber Overload](#)

[VelocityDb Namespace](#)

OidShort.ToString Method

Builds a string with PageNumber-SlotNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

PageNumber-SlotNumber

See Also



[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.OidShort Operators

The [OidShort](#) type exposes the following members.

Operators

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------|
|  | Equality | Compares id of two OidShort |
|  | Inequality | Compares id of two OidShort |

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Equality Operator

Compares id of two OidShort

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool operator ==(
    OidShort a,
    OidShort b
)
```

VB

```
Public Shared Operator = (
    a As OidShort,
    b As OidShort
) As Boolean
```

C++

```
public:
static bool operator ==(
    OidShort a,
    OidShort b
)
```

F#

```
static let inline (=)
    a : OidShort *
    b : OidShort : bool
```

Parameters

a

Type: [VelocityDb.OidShort](#)

first OidShort

b

Type: [VelocityDb.OidShort](#)

other OidShort

Return Value

Type: [Boolean](#)

true if id's are ==

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OidShort.Inequality Operator

Compares id of two OidShort

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public static bool operator !=(  
    OidShort a,  
    OidShort b  
)
```

```
VB  
Public Shared Operator <> (  
    a As OidShort,  
    b As OidShort  
) As Boolean
```

```
C++  
public:  
static bool operator !=(  
    OidShort a,  
    OidShort b  
)
```

```
F#  
static let inline (<>)  
    a : OidShort *  
    b : OidShort : bool
```

Parameters

a

Type: [VelocityDb.OidShort](#)

first OidShort

b

Type: [VelocityDb.OidShort](#)

other OidShort

Return Value

Type: [Boolean](#)

true if id's are !=

See Also

[OidShort Structure](#)

[VelocityDb Namespace](#)

OpFlags Enumeration

Flag bits for a persistent object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum OpFlags
```

VB

```
Public Enumeration OpFlags
```

C++

```
public enum class OpFlags
```

F#

```
type OpFlags
```

Members

| Member name | Value | Description |
|---------------------------|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IsUpdated | 1 | Object is updated |
| MembersLoaded | 2 | All members have been loaded |
| RemovedFromIndices | 4 | Call to Update() removed object from indices. If UpdateObject(IOptimizedPersistable, Boolean, Boolean) was used with 3rd parameter false, object is not removed from indices (flag value set accordingly) |

See Also

[VelocityDb Namespace](#)

OptimizedPersistable Class

Base class for all persistent capable classes except for embedded objects which do not need to be subclasses of this class.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.OptimizedPersistable

- [VelocityDb.Collection.BTree.BTreeByteArray](#)
- [VelocityDb.Collection.BTree.BTreeNode](#)
- [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)
- [VelocityDb.Collection.SortedSetBase\(Key\)](#)
- [VelocityDb.Collection.VelocityDbHashSet\(T\)](#)
- [VelocityDb.Collection.VelocityDbList\(T\)](#)
- [VelocityDb.Collection.WeakReferenceListBase\(T\)](#)
- [VelocityDb.Database](#)
- [VelocityDb.DatabaseLocation](#)
- [VelocityDb.DatabaseLocations](#)
- [VelocityDb.Indexing.IndexDescriptor](#)
- [VelocityDb.Indexing.Indexes](#)
- [VelocityDb.License](#)
- [VelocityDb.PageOffset](#)
- [VelocityDb.Placement](#)
- [VelocityDb.ReferenceTracked](#)
- [VelocityDb.Sync.Change](#)
- [VelocityDb.Sync.Changes](#)
- [VelocityDb.Sync.ReplicaSync](#)
- [VelocityDb.Sync.TransactionChanges](#)
- [VelocityDb.TypeInfo.DataMember](#)
- [VelocityDb.TypeInfo.Reference](#)
- [VelocityDb.TypeInfo.Relation](#)
- [VelocityDb.TypeInfo.Schema](#)
- [VelocityDb.TypeInfo.TypeVersion](#)
- [VelocityDb.TypeInfo.VelocityDbType](#)
- [VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray](#)
- [VelocityDBExtensions.CompressedBitArray.RunningLengthWord](#)
- [VelocityDBExtensions.Spatial.RTree](#)
- [VelocityDBExtensions2.AspNet.Identity.AspNetIdentity](#)
- [VelocityDBExtensions2.AspNet.Identity.IdentityRole](#)
- [VelocityDBExtensions2.AspNet.Identity.IdentityUser](#)
- [VelocityDBExtensions2.AspNet.Identity.UserLoginInfoAdapter](#)
- [VelocityGraph.EdgeType](#)
- [VelocityGraph.Graph](#)
- [VelocityGraph.PropertyType](#)

[VelocityGraph.VertexType](#)Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class OptimizedPersistable : IOptimizedPersistable,
    IComparable, ICloneable
```

VB

```
<SerializableAttribute>
Public Class OptimizedPersistable
    Implements IOptimizedPersistable, IComparable, ICloneable
```

C++



```
[SerializableAttribute]
public ref class OptimizedPersistable : IOptimizedPersistable,
    IComparable, ICloneable
```

F#


```
[<SerializableAttribute>]
type OptimizedPersistable =
    class
        interface IOptimizedPersistable
        interface IComparable
        interface ICloneable
    end
```















The **OptimizedPersistable** type exposes the following members.





Constructors

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OptimizedPersistable() | The default constructor can be used to create simple OptimizedPersistable objects. |
|  OptimizedPersistable(UInt64) | This constructor can be used in cases where the object id is known but you don't want to open the object, maybe you just want to add this object to a BTreeSet. |














Properties



















| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override |






| | | |
|-------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | this to return false for types that should not share pages with other types. |
|  | Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. |
|  | DatabaseNumber | Gets the database number of this object. |
|  | FieldsLoaded | Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an overrode of LazyLoadFields returns true then the object members are loaded by calling <code>[:OptimizedPersistable.LoadFields]</code> |
|  | FlushIfPageFull | By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page) |
|  | Id | Gets the id of this object. The id is structured as Oid |
|  | IsPersistent | Gets the persistent state of an object. An object is considered persistent when it has an Oid , that is the Id is not 0 |
|  | IsUpdated | Gets the updated state of the object |
|  | LazyLoadFields | By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading). |
|  | MaxNumberOfDatabases | Possibly restrict instances of to a single Database. By default this property is <code>UInt32.MaxValue</code> but classes like <code>BTreeSetOidShort</code> , <code>BTreeMapShortOid</code> ... override this property to return 1 since short references are restricted to a single Database. |
|  | ObjectsPerPage | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  | Oid | Gets the object identifier represented as an Oid . |
|  | OidShort | Gets the OidShort part of the object Oid |
|  | PageNumber | Gets the page number part of the object Oid (Id) |
|  | PagesPerDatabase | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability |

| | | |
|-----------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  | PlacementDatabaseNumber | Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given. |
|  | RemovedFromIndices | <code>true</code> if call to [!:Update(bool)] or UpdateObject(IOptimizedPersistable, Boolean, Boolean) caused this object to be removed from possible indices |
|  | ShortId | Gets the short id of this object. The id is structured as OidShort |
|  | SlotNumber | Gets the page number from the object identifier |



Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Clone | See MemberwiseClone() |
|  | CompareTo | Compares objects using the object Id |
|  | Equals | Determines whether the specified Object is equal to the current Object. If this object or other object isn't persistent we call Equals(Object) (Overrides Object.Equals(Object) .) |
|  | FlushTransients | Process data in transient fields (if any), do whatever is required to persist such data if needed. By default does nothing. |
|  | GetDataMembers | Gets a list of the fields meta data objects |
|  | GetFieldValues | Returns a list of field values of this object or returns it as an array (if this is an array) |
|  | GetHashCode | Computes a hash code based on Oid of object. Make sure to persist objects before using GetHashCode() or override GetHashCode() (Overrides Object.GetHashCode() .) |
|  | GetPage | Get the persistent storage Page of this object |
|  | GetSession | Gets the session of this object or null if this object isn't yet persisted. |
|  | GetTypeVersion | The database engine needs this internally |
|  | GetWrappedObject | Internally used when IOptimizedPersistable is a wrapper for a non IOptimizedPersistable |
|  | InitializeAfterRead | This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data. |
|  | InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully |

| | | |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | loaded. Override this to provide your own initializations of transient data. |
|  | ListToString | This is a support function for the VelocityDbBrowser |
|  | OidOf | Gets the Oid (encoded as an UInt64) of a non OptimizedPersistable object or 0 if the object isn't persistent on the same page as this object. |
|  | Open(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |
|  | Open(T)(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |
|  | OptimizedPersistableFieldValues | Gets a list of all field values of the current object that are subclasses of OptimizedPersistable |
|  | Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
|  | Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. |
|  | PersistentVersion | Gets the page version of a persistent object |
|  | PersistMyReferences | Persists references from this object |
|  | ReadMe | Used by code generator |
|  | ReadMeUsingSchemaReflection | Used by code generator. By default objects are read and written using persisted schema information and reflection. |
|  | SetPage | Sets the persistent storage Page of this object |
|  | SetTypeVersion | The database engine needs this internally |
|  | ShallowCopyTo | Internal use for now |
|  | ToString | Displays class name plus object id (Overrides Object.ToString() .) |
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. |
|  | Update() | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! |
|  | Update(Action) | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! Add code to update object in Action parameter. |

| | |
|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  UpdateNonIndexField | Marks an object as being updated so that object will be written at commit transaction. Call before updating a NON indexed field. Same as Update() but avoids deleting object from indices (if any). If updating a field used in any index, call Update() instead or you may end up with one or more corrupt indices. |
|  UpdateTypeVersion | Updates the object and make this object use the latest Type definition for its class. The object now will now adjust for data attribute changes. |
|  Write | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page will also be written. |
|  WriteMe | Used by code generator. |
|  WriteMeUsingSchemaReflection | Used by code generator. This is the default way of writing objects using schema information and .NET reflection info. |

Extension Methods



| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb Namespace](#)

OptimizedPersistable Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | OptimizedPersistable() | The default constructor can be used to create simple OptimizedPersistable objects. |
|  | OptimizedPersistable(UInt64) | This constructor can be used in cases where the object id is known but you don't want to open the object, maybe you just want to add this object to a BTreeSet. |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable Constructor

The default constructor can be used to create simple OptimizedPersistable objects.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OptimizedPersistable ()
```

VB

```
Public Sub New
```

C++

```
public:  
OptimizedPersistable ()
```

F#

```
new : unit -> OptimizedPersistable
```

See Also

[OptimizedPersistable Class](#)

[OptimizedPersistable Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable Constructor (UInt64)

This constructor can be used in cases where the object id is known but you don't want to open the object, maybe you just want to add this object to a BTreeSet.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OptimizedPersistable(  
    ulong id  
)
```

VB

```
Public Sub New (  
    id As ULong  
)
```

C++

```
public:  
OptimizedPersistable(  
    unsigned long long id  
)
```

F#

```
new :  
    id : uint64 -> OptimizedPersistable
```

Parameters

id

Type: [System.UInt64](#)

Object id to use for this object

See Also

[OptimizedPersistable Class](#)












[OptimizedPersistable Overload](#)









[VelocityDb Namespace](#)

OptimizedPersistable.OptimizedPersistable Properties

The [OptimizedPersistable](#) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. |
|  DatabaseNumber | Gets the database number of this object. |
|  FieldsLoaded | Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an overrode of LazyLoadFields returns true then the object members are loaded by calling [!:OptimizedPersistable.LoadFields] |
|  FlushIfPageFull | By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page) |
|  Id | Gets the id of this object. The id is structured as Oid |
|  IsPersistent | Gets the persistent state of an object. An object is considered persistent when it has an Oid , that is the Id is not 0 |
|  IsUpdated | Gets the updated state of the object |
|  LazyLoadFields | By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading). |
|  MaxNumberOfDatabases | Possibly restrict instances of to a single Database. By default this property is UInt32.MaxValue but classes like BTreeSetOidShort, BTreeMapShortOid ... override this property to return 1 since short references are restricted to a single Database. |
|  ObjectsPerPage | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |

| | |
|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Oid | Gets the object identifier represented as an Oid . |
|  OidShort | Gets the OidShort part of the object Oid |
|  PageNumber | Gets the page number part of the object Oid (Id) |
|  PagesPerDatabase | A default for number of objects per database page used when persisting objects without an explicit Placement object or if persisted using Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent. |
|  PlacementDatabaseNumber | Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given. |
|  RemovedFromIndices | true if call to [!:Update(bool)] or UpdateObject(IOptimizedPersistable, Boolean, Boolean) caused this object to be removed from possible indices |
|  ShortId | Gets the short id of this object. The id is structured as OidShort |
|  SlotNumber | Gets the page number from the object identifier |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overridable ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get ();  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Cache Property

By default, the an object cache is determined by a [SessionBase](#) constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual CacheEnum Cache { get; }
```

VB

```
Public Overridable ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get ();  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.DatabaseNumber Property

Gets the database number of this object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint DatabaseNumber { get; }
```

VB

```
Public ReadOnly Property DatabaseNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int DatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
member DatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

The database number of this object or 0 if object is not persistent

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.FieldsLoaded Property

Gets the field members load state. By default, all members are loaded when a persistent object is opened but if an override of [LazyLoadFields](#) returns true then the object members are loaded by calling **[!:OptimizedPersistable.LoadFields]**

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool FieldsLoaded { get; set; }
```

VB

```
Public Property FieldsLoaded As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool FieldsLoaded {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

F#

```
abstract FieldsLoaded : bool with get, set  
override FieldsLoaded : bool with get, set
```

Property Value

Type: [Boolean](#)

true if all members are loaded; otherwise false

Implements

[IOptimizedPersistable.FieldsLoaded](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.FlushIfPageFull Property

By default we flush (write) any updated page we find when looking for an object placement page and the page is considered full (depends on how many objects we permit/page)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool FlushIfPageFull { get; }
```

VB

```
Public Overridable ReadOnly Property FlushIfPageFull As Boolean  
    Get
```

C++

```
public:  
virtual property bool FlushIfPageFull {  
    bool get ();  
}
```

F#

```
abstract FlushIfPageFull : bool with get  
override FlushIfPageFull : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.FlushIfPageFull](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Id Property

Gets the id of this object. The id is structured as [Oid](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Id { get; set; }
```

VB

```
Public Property Id As ULong  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned long long Id {  
    unsigned long long get () sealed;  
    void set (unsigned long long value) sealed;  
}
```

F#

```
abstract Id : uint64 with get, set  
override Id : uint64 with get, set
```

Property Value

Type: [UInt64](#)

The id of this object or 0 if object is not persistent

Implements

[IOptimizedPersistable.Id](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.IsPersistent Property

Gets the persistent state of an object. An object is considered persistent when it has an [Oid](#), that is the [Id](#) is not 0

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsPersistent { get; }
```

VB

```
Public ReadOnly Property IsPersistent As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsPersistent {  
    bool get () sealed;  
}
```

F#

```
abstract IsPersistent : bool with get  
override IsPersistent : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.IsPersistent](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.IsUpdated Property

Gets the updated state of the object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool IsUpdated { get; set; }
```

VB

```
Public Overridable Property IsUpdated As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool IsUpdated {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract IsUpdated : bool with get, set  
override IsUpdated : bool with get, set
```

Property Value

Type: [Boolean](#)

true if updated; otherwise false

Implements

[IOptimizedPersistable.IsUpdated](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.LazyLoadFields Property

By default all fields are loaded when opening a persistent object but an option is provided to load members on demand (lazy loading).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool LazyLoadFields { get; }
```

VB

```
Public Overridable ReadOnly Property LazyLoadFields As Boolean  
    Get
```

C++

```
public:  
virtual property bool LazyLoadFields {  
    bool get ();  
}
```

F#

```
abstract LazyLoadFields : bool with get  
override LazyLoadFields : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.LazyLoadFields](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.MaxNumberOfDatabases Property

Possibly restrict instances to a single Database. By default this property is UInt32.MaxValue but classes like BTreeSetOidShort, BTreeMapShortOid ... override this property to return 1 since short references are restricted to a single Database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overridable ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get ();  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ObjectsPerPage Property

A default for number of objects per database page used when persisting objects without an explicit [Placement](#) object or if persisted using [Persist\(SessionBase, IOptimizedPersistable, Boolean, Boolean\)](#). This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ushort ObjectsPerPage { get; }
```

VB

```
Public Overridable ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get ();  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Return Value

Type: [UInt16](#)

The requested number of objects per page.

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Oid Property

Gets the object identifier represented as an **Oid**.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid Oid { get; }
```

VB

```
Public ReadOnly Property Oid As Oid  
    Get
```

C++

```
public:  
property Oid Oid {  
    Oid get ();  
}
```

F#

```
member Oid : Oid with get
```

Property Value

Type: [Oid](#)

The **Oid** of the object

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.OidShort Property

Gets the OidShort part of the object Oid

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OidShort OidShort { get; }
```

VB

```
Public ReadOnly Property OidShort As OidShort  
    Get
```

C++

```
public:  
property OidShort OidShort {  
    OidShort get ();  
}
```

F#

```
member OidShort : OidShort with get
```

Property Value

Type: [OidShort](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.PageNumber Property

Gets the page number part of the object Oid (Id)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort PageNumber { get; }
```

VB

```
Public ReadOnly Property PageNumber As UShort  
    Get
```

C++

```
public:  
property unsigned short PageNumber {  
    unsigned short get ();  
}
```

F#

```
member PageNumber : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.PagesPerDatabase Property

A default for number of objects per database page used when persisting objects without an explicit [Placement](#) object or if persisted using [Persist\(SessionBase, IOptimizedPersistable, Boolean, Boolean\)](#). This happens when objects are persisted by reachability from a persistent object. All objects reachable from a persistent object are automatically made persistent.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ushort PagesPerDatabase { get; }
```

VB

```
Public Overridable ReadOnly Property PagesPerDatabase As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short PagesPerDatabase {  
    unsigned short get ();  
}
```

F#

```
abstract PagesPerDatabase : uint16 with get  
override PagesPerDatabase : uint16 with get
```

Return Value

Type: [UInt16](#)

The requested number of pages per database

Implements

[IOptimizedPersistable.PagesPerDatabase](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.PlacementDatabaseNumber Property

Gets the Database Id number to use when placing (persisting) an instance of this class when no other placement directive has been given.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual uint PlacementDatabaseNumber { get; }
```

VB

```
Public Overridable ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int PlacementDatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get  
override PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.PlacementDatabaseNumber](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.RemovedFromIndices Property

true

if call to **[!:Update(bool)]** or [UpdateObject\(IOptimizedPersistable, Boolean, Boolean\)](#) caused this object to be removed from possible indices

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool RemovedFromIndices { get; set; }
```

VB

```
Public Property RemovedFromIndices As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool RemovedFromIndices {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

F#

```
abstract RemovedFromIndices : bool with get, set  
override RemovedFromIndices : bool with get, set
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.RemovedFromIndices](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ShortId Property

Gets the short id of this object. The id is structured as [OidShort](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ShortId { get; }
```

VB

```
Public ReadOnly Property ShortId As UInteger  
    Get
```

C++

```
public:  
property unsigned int ShortId {  
    unsigned int get ();  
}
```

F#

```
member ShortId : uint32 with get
```

Property Value

Type: [UInt32](#)

The short id of this object or 0 if object is not persistent

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.SlotNumber Property

Gets the page number from the object identifier

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort SlotNumber { get; }
```

VB

```
Public ReadOnly Property SlotNumber As UShort  
    Get
```

C++

```
public:  
property unsigned short SlotNumber {  
    unsigned short get ();  
}
```

F#

```
member SlotNumber : uint16 with get
```

Property Value

Type: [UInt16](#)

the page number of this object

See Also

















[OptimizedPersistable Class](#)



















[VelocityDb Namespace](#)




OptimizedPersistable.OptimizedPersistable Methods

The [OptimizedPersistable](#) type exposes the following members.



Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Clone | See MemberwiseClone() |
|  CompareTo | Compares objects using the object Id |
|  Equals | Determines whether the specified Object is equal to the current Object. If this object or other object isn't persistent we call Equals(Object) (Overrides Object.Equals(Object) .) |
|  FlushTransients | Process data in transient fields (if any), do whatever is required to persist such data if needed. By default does nothing. |
|  GetDataMembers | Gets a list of the fields meta data objects |
|  GetFieldValues | Returns a list of field values of this object or returns it as an array (if this is an array) |
|  GetHashCode | Computes a hash code based on Oid of object. Make sure to persist objects before using GetHashCode() or override GetHashCode() (Overrides Object.GetHashCode() .) |
|  GetPage | Get the persistent storage Page of this object |
|  GetSession | Gets the session of this object or null if this object isn't yet persisted. |
|  GetTypeVersion | The database engine needs this internally |
|  GetWrappedObject | Internally used when IOptimizedPersistable is a wrapper for a non IOptimizedPersistable |
|  InitializeAfterRead | This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data. |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. |
|  ListToString | This is a support function for the VelocityDbBrowser |
|  OidOf | Gets the Oid (encoded as an UInt64) of a non OptimizedPersistable object or 0 if the object isn't persistent on the same page as this object. |
|  Open(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Open(T)(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |
|  OptimizedPersistableFieldValues | Gets a list of all field values of the current object that are subclasses of OptimizedPersistable |
|  Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
|  Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. |
|  PersistentVersion | Gets the page version of a persistent object |
|  PersistMyReferences | Persists references from this object |
|  ReadMe | Used by code generator |
|  ReadMeUsingSchemaReflection | Used by code generator. By default objects are read and written using persisted schema information and reflection. |
|  SetPage | Sets the persistent storage Page of this object |
|  SetTypeVersion | The database engine needs this internally |
|  ShallowCopyTo | Internal use for now |
|  ToString | Displays class name plus object id (Overrides Object.ToString() .) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. |
|  Update() | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! |
|  Update(Action) | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! Add code to update object in Action parameter. |
|  UpdateNonIndexField | Marks an object as being updated so that object will be written at commit transaction. Call before updating a NON indexed field. Same as Update() but avoids deleting object from indices (if any). If updating a field used in any index, call Update() instead or you may end up with one or more corrupt indices. |
|  UpdateTypeVersion | Updates the object and make this object use the latest Type definition for its class. The object now will now adjust for data attribute changes. |
|  Write | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| | | updated. Other objects on the same page will also be written. |
|  | WriteMe | Used by code generator. |
|   | WriteMeUsingSchemaReflection | Used by code generator. This is the default way of writing objects using schema information and .NET reflection info. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Clone Method

See [MemberwiseClone\(\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Object Clone ()
```

VB

```
Public Overridable Function Clone As Object
```

C++

```
public:  
virtual Object^ Clone ()
```

F#

```
abstract Clone : unit -> Object  
override Clone : unit -> Object
```

Return Value

Type: [Object](#)

A non persisted clone

Implements

[ICloneable.Clone\(\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.CompareTo Method

Compares objects using the object [Id](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual int CompareTo(  
    Object obj  
)
```

VB

```
Public Overridable Function CompareTo (  
    obj As Object  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Object^ obj  
)
```

F#

```
abstract CompareTo :  
    obj : Object -> int  
override CompareTo :  
    obj : Object -> int
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for "M:VelocityDb.OptimizedPersistable.CompareTo(System.Object)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.CompareTo(System.Object)"]

Implements

[IComparable.CompareTo\(Object\)](#)

See Also

[OptimizedPersistable Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

OptimizedPersistable.Equals Method

Determines whether the specified Object is equal to the current Object. If this object or other object isn't persistent we call [Equals\(Object\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

Object to compare with

Field Value

Type: [Boolean](#)

true if the specified Object is equal to the current Object; otherwise, false.

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.Equals(System.Object)"]

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.FlushTransients Method

Process data in transient fields (if any), do whatever is required to persist such data if needed. By default does nothing.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void FlushTransients ()
```

VB

```
Public Overridable Sub FlushTransients
```

C++

```
public:  
virtual void FlushTransients ()
```

F#

```
abstract FlushTransients : unit -> unit  
override FlushTransients : unit -> unit
```

Implements

[IOptimizedPersistable.FlushTransients\(\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetDataMembers Method

Gets a list of the fields meta data objects

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<DataMember> GetDataMembers ()
```

VB

```
Public Function GetDataMembers As List(Of DataMember)
```

C++

```
public:  
List<DataMember^>^ GetDataMembers ()
```

F#

```
member GetDataMembers : unit -> List<DataMember>
```

Return Value

Type: [List\(DataMember\)](#)

List of DataMember

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetFieldValues Method

Returns a list of field values of this object or returns it as an array (if this is an array)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<Object> GetFieldValues ()
```

VB

```
Public Function GetFieldValues As List(Of Object)
```

C++

```
public:  
List<Object^> GetFieldValues ()
```

F#

```
member GetFieldValues : unit -> List<Object>
```

Return Value

Type: [List\(Object\)](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.GetFieldValues"]

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetHashCode Method

Computes a hash code based on Oid of object. Make sure to persist objects before using GetHashCode() or override GetHashCode()

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode ()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode () override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.GetHashCode"]

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetPage Method

Get the persistent storage [Page](#) of this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Page GetPage ()
```

VB

```
Public Function GetPage As Page
```

C++

```
public:  
virtual Page^ GetPage () sealed
```

F#

```
abstract GetPage : unit -> Page  
override GetPage : unit -> Page
```

Return Value

Type: [Page](#)

The persistent storage [Page](#) of this object

Implements

[IOptimizedPersistable.GetPage\(\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetSession Method

Gets the session of this object or null if this object isn't yet persisted.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionBase GetSession()
```

VB

```
Public Function GetSession As SessionBase
```

C++

```
public:  
SessionBase^ GetSession()
```

F#

```
member GetSession : unit -> SessionBase
```

Return Value

Type: [SessionBase](#)

The active [SessionBase](#) of this object

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetTypeVersion Method

The database engine needs this internally

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion GetTypeVersion()
```

VB

```
Public Function GetTypeVersion As TypeVersion
```

C++

```
public:  
virtual TypeVersion^ GetTypeVersion() sealed
```

F#

```
abstract GetTypeVersion : unit -> TypeVersion  
override GetTypeVersion : unit -> TypeVersion
```

Return Value

Type: [TypeVersion](#)

The [TypeVersion](#) of this object

Implements

[IOptimizedPersistable.GetTypeVersion\(\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.GetWrappedObject Method

Internally used when [IOptimizedPersistable](#) is a wrapper for a non [IOptimizedPersistable](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetWrappedObject ()
```

VB

```
Public Function GetWrappedObject As Object
```

C++

```
public:  
virtual Object^ GetWrappedObject () sealed
```

F#

```
abstract GetWrappedObject : unit -> Object  
override GetWrappedObject : unit -> Object
```

Return Value

Type: [Object](#)

Wrapped object if set; otherwise this object

Implements

[IOptimizedPersistable.GetWrappedObject\(\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.InitializeAfterRead Method

This function is called when an object has been read from disk and all data members (fields) have been loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overridable Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
)
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overridable Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
)
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ListToString Method

This is a support function for the VelocityDbBrowser

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string ListToString(  
    IList list,  
    Page page  
)
```

VB

```
Public Shared Function ListToString (  
    list As IList,  
    page As Page  
) As String
```

C++

```
public:  
static String^ ListToString(  
    IList^ list,  
    Page^ page  
)
```

F#

```
static member ListToString :  
    list : IList *  
    page : Page -> string
```

Parameters

list

Type: [System.Collections.IList](#)

A list to output as a string

page

Type: [VelocityDb.Page](#)

The page containing the list

Return Value

Type: [String](#)

A string representing the input list.

See Also

[OptimizedPersistable Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

OptimizedPersistable.OidOf Method

Gets the Oid (encoded as an UInt64) of a non OptimizedPersistable object or 0 if the object isn't persistent on the same page as this object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong OidOf(  
    Object obj  
)
```

VB

```
Public Function OidOf (  
    obj As Object  
) As ULong
```

C++

```
public:  
unsigned long long OidOf(  
    Object^ obj  
)
```

F#

```
member OidOf :  
    obj : Object -> uint64
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for "M:VelocityDb.OptimizedPersistable.OidOf(System.Object)"]

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.OidOf(System.Object)"]



See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Open Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------|-----------------------------------------------------------------------|
|  | Open(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |
|  | Open(T)(UInt32, Boolean, Boolean) | Opens a persistent object located in the same Database as this object |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Open Method (UInt32, Boolean, Boolean)

Opens a persistent object located in the same Database as this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object Open(  
    uint pageSlot,  
    bool update = false,  
    bool inFlush = false  
)
```

VB

```
Public Function Open (  
    pageSlot As UInteger,  
    Optional update As Boolean = false,  
    Optional inFlush As Boolean = false  
) As Object
```

C++

```
public:  
Object^ Open(  
    unsigned int pageSlot,  
    bool update = false,  
    bool inFlush = false  
)
```

F#

```
member Open :  
    pageSlot : uint32 *  
    ?update : bool *  
    ?inFlush : bool  
(* Defaults:  
    let_update = defaultArg update false  
    let_inFlush = defaultArg inFlush false  
)  
-> Object
```

Parameters

pageSlot

Type: [System.UInt32](#)

The ObjectId encoded as a UInt32 of the object to open within the same Database as this object

update (Optional)

Type: [System.Boolean](#)

VelocityDB Class Library

Open the object for update?

inFlush (Optional)

Type: [System.Boolean](#)

if true, disallow page flushing while opening the object

Return Value

Type: [Object](#)

The opened object or `null` if it does not exist

See Also

[OptimizedPersistable Class](#)

[Open Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Open(T) Method (UInt32, Boolean, Boolean)

Opens a persistent object located in the same Database as this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T Open<T>(
    uint pageSlot,
    bool update = false,
    bool inFlush = false
)
```

VB

```
Public Function Open(Of T) (
    pageSlot As UInteger,
    Optional update As Boolean = false,
    Optional inFlush As Boolean = false
) As T
```

C++

```
public:
    generic<typename T>
    T Open(
        unsigned int pageSlot,
        bool update = false,
        bool inFlush = false
    )
```

F#

```
member Open :
    pageSlot : uint32 *
    ?update : bool *
    ?inFlush : bool
(* Defaults:
    let _update = defaultArg update false
    let _inFlush = defaultArg inFlush false
*)
-> 'T
```

Parameters

pageSlot

Type: [System.UInt32](#)

The OidShort encoded as a UInt32 of the object to open within the same Database as this object

update (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Open the object for update?

inFlush (Optional)

Type: [System.Boolean](#)

if true, disallow page flushing while opening the object

Type Parameters

T

**[Missing <typeparam name="T"/> documentation for
"M:VelocityDb.OptimizedPersistable.Open`1(System.UInt32,System.Boolean,System.Boolean)"]**

Return Value

Type: **T**

The opened object or `null` if it does not exist

See Also

[OptimizedPersistable Class](#)

[Open Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.OptimizedPersistableFieldValues Method

Gets a list of all field values of the current object that are subclasses of OptimizedPersistable

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<OptimizedPersistable> OptimizedPersistableFieldValues ()
```

VB

```
Public Function OptimizedPersistableFieldValues As List(Of  
OptimizedPersistable)
```

C++

```
public:  
List<OptimizedPersistable^> OptimizedPersistableFieldValues ()
```

F#

```
member OptimizedPersistableFieldValues : unit -> List<OptimizedPersistable>
```

Return Value

Type: [List\(OptimizedPersistable\)](#)

List of OptimizedPersistable



See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Persist Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|  | Persist(SessionBase, IOptimizedPersistable, Boolean, Boolean) | Persists this object. |
|  | Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Persist Method (SessionBase, IOptimizedPersistable, Boolean, Boolean)

Persists this object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ulong Persist(
    SessionBase session,
    IOptimizedPersistable placeHint,
    bool persistRefs = true,
    bool disableFlush = false
)
```

VB

```
Public Overridable Function Persist (
    session As SessionBase,
    placeHint As IOptimizedPersistable,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    SessionBase^ session,
    IOptimizedPersistable^ placeHint,
    bool persistRefs = true,
    bool disableFlush = false
)
```

F#

```
abstract Persist :
    session : SessionBase *
    placeHint : IOptimizedPersistable *
    ?persistRefs : bool *
    ?disableFlush : bool
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
*)
-> uint64
override Persist :
    session : SessionBase *
    placeHint : IOptimizedPersistable *
    ?persistRefs : bool *
    ?disableFlush : bool
```

```
(* Defaults:  
    let_persistRefs = defaultArg persistRefs true  
    let_disableFlush = defaultArg disableFlush false  
*)  
-> uint64
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

placeHint

Type: [VelocityDb.IOptimizedPersistable](#)

Use placement as specified by this object type, see [PlacementDatabaseNumber](#), [ObjectsPerPage](#) and [PagesPerDatabase](#) For best performance this should be a recently created object that you want the current object to be persisted near.

persistRefs (Optional)

Type: [System.Boolean](#)

Persist any referenced object now or delay until flush/commit

disableFlush (Optional)

Type: [System.Boolean](#)

Controls possible flushing of updated pages. Set to true if you want to prevent updated pages from being flushed to disk and setting such pages to a non updated state.

Return Value

Type: [UInt64](#)

The object id of the persistent object

Implements

[IOptimizedPersistable.Persist\(SessionBase, IOptimizedPersistable, Boolean, Boolean\)](#)

See Also

[OptimizedPersistable Class](#)

[Persist Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Persist Method (Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable))

Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overridable Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
)
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
```

```
-> uint64
override Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of this persistent object

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue<IOptimizedPersistable>\)](#)

See Also

[OptimizedPersistable Class](#)

[Persist Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.PersistentVersion Method

Gets the page version of a persistent object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong PersistentVersion()
```

VB

```
Public Function PersistentVersion As ULong
```

C++

```
public:  
unsigned long long PersistentVersion()
```

F#

```
member PersistentVersion : unit -> uint64
```

Return Value

Type: [UInt64](#)

The page version.

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.PersistMyReferences Method

Persists references from this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void PersistMyReferences (  
    SessionBase session,  
    bool inFlush  
)
```

VB

```
Public Overridable Sub PersistMyReferences (  
    session As SessionBase,  
    inFlush As Boolean  
)
```

C++

```
public:  
virtual void PersistMyReferences (  
    SessionBase^ session,  
    bool inFlush  
)
```

F#

```
abstract PersistMyReferences :  
    session : SessionBase *  
    inFlush : bool -> unit  
override PersistMyReferences :  
    session : SessionBase *  
    inFlush : bool -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

inFlush

Type: [System.Boolean](#)

are we in a page flush

Implements

[IOptimizedPersistable.PersistMyReferences\(SessionBase, Boolean\)](#)

VelocityDB Class Library

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ReadMe Method

Used by code generator

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ReadMe(  
    TypeVersion typeVersion,  
    byte[] memberBytes,  
    ref int offset,  
    SessionBase session,  
    Page page,  
    bool useOidShort,  
    Schema schema,  
    bool openRefs,  
    List<IOptimizedPersistable> toLoadMembers,  
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
)
```

VB

```
Public Overridable Sub ReadMe (  
    typeVersion As TypeVersion,  
    memberBytes As Byte(),  
    ByRef offset As Integer,  
    session As SessionBase,  
    page As Page,  
    useOidShort As Boolean,  
    schema As Schema,  
    openRefs As Boolean,  
    toLoadMembers As List(Of IOptimizedPersistable),  
    graphDepth As Integer,  
    graphDepthToLoad As Integer,  
    primitivesOnly As Boolean  
)
```

C++

```
public:  
virtual void ReadMe(  
    TypeVersion^ typeVersion,  
    array<unsigned char>^ memberBytes,  
    int% offset,  
    SessionBase^ session,  
    Page^ page,  
    bool useOidShort,  
    Schema^ schema,  
    bool openRefs,  
    List<IOptimizedPersistable^>^ toLoadMembers,
```

```
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
)
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema

a, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System.Int32@, VelocityDb.Session.SessionBase, VelocityDb.Page, System.Boolean, VelocityDb.TypeInfo.Schema, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System.Int32@, VelocityDb.Session.SessionBase, VelocityDb.Page, System.Boolean, VelocityDb.TypeInfo.Schema, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System.Int32@, VelocityDb.Session.SessionBase, VelocityDb.Page, System.Boolean, VelocityDb.TypeInfo.Schema, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System.Int32@, VelocityDb.Session.SessionBase, VelocityDb.Page, System.Boolean, VelocityDb.TypeInfo.Schema, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System.Int32@, VelocityDb.Session.SessionBase, VelocityDb.Page, System.Boolean, VelocityDb.TypeInfo.Schema, System.Boolean, System.Collections.Generic.List{VelocityDb.IOptimizedPersistable}, System.Int32, System.Int32, System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List{IOptimizedPersistable}](#)

[Missing <param name="toLoadMembers"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion, System.Byte[], System

`.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]`

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.OptimizedPersistable.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List\(IOptimizedPersistable\), Int32, Int32, Boolean\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ReadMeUsingSchemaReflection Method

Used by code generator. By default objects are read and written using persisted schema information and reflection.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ReadMeUsingSchemaReflection(
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    IOptimizedPersistable pObj,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Shared Sub ReadMeUsingSchemaReflection (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    pObj As IOptimizedPersistable,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
    static void ReadMeUsingSchemaReflection(
        TypeVersion^ typeVersion,
        array<unsigned char>^ memberBytes,
        int% offset,
        IOptimizedPersistable^ pObj,
        SessionBase^ session,
        Page^ page,
```

```
bool useOidShort,  
Schema^ schema,  
bool openRefs,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
)
```

F#

```
static member ReadMeUsingSchemaReflection :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    pObj : IOptimizedPersistable *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

Version of a type

memberBytes

Type: [System.Byte\[\]](#)

Bytes containing object to read

offset

Type: [System.Int32](#)

Offset into memberBytes

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object being read in

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

page

Type: [VelocityDb.Page](#)

Page of object being read

VelocityDB Class Library

useOidShort

Type: [System.Boolean](#)

Does object use short object references (32bit)?

schema

Type: [VelocityDb.TypeInfo.Schema](#)

Schema information

openRefs

Type: [System.Boolean](#)

Shall references from object also be filled in now

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

Members remaining to be read

graphDepth

Type: [System.Int32](#)

Current graph depth read

graphDepthToLoad

Type: [System.Int32](#)

Stop reading at this graph depth

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for

"M:VelocityDb.OptimizedPersistable.ReadMeUsingSchemaReflection(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.SetPage Method

Sets the persistent storage [Page](#) of this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetPage (  
    Page page  
)
```

VB

```
Public Sub SetPage (  
    page As Page  
)
```

C++

```
public:  
virtual void SetPage (  
    Page^ page  
) sealed
```

F#

```
abstract SetPage :  
    page : Page -> unit  
override SetPage :  
    page : Page -> unit
```

Parameters

page

Type: [VelocityDb.Page](#)

A persistent storage [Page](#)

Implements

[IOptimizedPersistable.SetPage\(Page\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.SetTypeVersion Method

The database engine needs this internally

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetTypeVersion(  
    TypeVersion typeVersion  
)
```

VB

```
Public Sub SetTypeVersion (  
    typeVersion As TypeVersion  
)
```

C++

```
public:  
virtual void SetTypeVersion(  
    TypeVersion^ typeVersion  
) sealed
```

F#

```
abstract SetTypeVersion :  
    typeVersion : TypeVersion -> unit  
override SetTypeVersion :  
    typeVersion : TypeVersion -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for "M:VelocityDb.OptimizedPersistable.SetTypeVersion(VelocityDb.TypeInfo.TypeVersion)"]

Implements

[IOptimizedPersistable.SetTypeVersion\(TypeVersion\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.ShallowCopyTo Method

Internal use for now

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IOptimizedPersistable ShallowCopyTo(  
    Page page  
)
```

VB

```
Public Function ShallowCopyTo (  
    page As Page  
) As IOptimizedPersistable
```

C++

```
public:  
virtual IOptimizedPersistable^ ShallowCopyTo(  
    Page^ page  
) sealed
```

F#

```
abstract ShallowCopyTo :  
    page : Page -> IOptimizedPersistable  
override ShallowCopyTo :  
    page : Page -> IOptimizedPersistable
```

Parameters

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.OptimizedPersistable.ShallowCopyTo(VelocityDb.Page)"]

Return Value

Type: [IOptimizedPersistable](#)

[Missing <returns> documentation for "M:VelocityDb.OptimizedPersistable.ShallowCopyTo(VelocityDb.Page)"]

Implements

[IOptimizedPersistable.ShallowCopyTo\(Page\)](#)

See Also

[OptimizedPersistable Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

OptimizedPersistable.ToString Method

Displays class name plus object id

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overridable Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
)
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)



See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Update Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Update() | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! |
|  | Update(Action) | Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! Add code to update object in Action parameter. |

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Update Method

Marks an object as being updated so that object will be written at commit transaction. Call before making object changes!

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Update ()
```

VB

```
Public Function Update As Boolean
```

C++

```
public:  
virtual bool Update () sealed
```

F#

```
abstract Update : unit -> bool  
override Update : unit -> bool
```

Return Value

Type: [Boolean](#)

true if update successful

Implements

[IOptimizedPersistable.Update\(\)](#)

See Also

[OptimizedPersistable Class](#)

[Update Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Update Method (Action)

Marks an object as being updated so that object will be written at commit transaction. Call before making object changes! Add code to update object in [Action](#) parameter.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Update(  
    Action codeBlock  
)
```

VB

```
Public Function Update (  
    codeBlock As Action  
) As Boolean
```

C++

```
public:  
bool Update(  
    Action^ codeBlock  
)
```

F#

```
member Update :  
    codeBlock : Action -> bool
```

Parameters

codeBlock

Type: [System.Action](#)

Block of code doing actual update to the object

Return Value

Type: [Boolean](#)

true if update successful

See Also

[OptimizedPersistable Class](#)

[Update Overload](#)

[VelocityDb Namespace](#)

OptimizedPersistable.UpdateNonIndexField Method

Marks an object as being updated so that object will be written at commit transaction. Call before updating a NON indexed field. Same as [Update\(\)](#) but avoids deleting object from indices (if any). If updating a field used in any index, call [Update\(\)](#) instead or you may end up with one or more corrupt indices.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool UpdateNonIndexField()
```

VB

```
Public Function UpdateNonIndexField As Boolean
```

C++

```
public:  
bool UpdateNonIndexField()
```

F#

```
member UpdateNonIndexField : unit -> bool
```

Return Value

Type: [Boolean](#)

true if update successful

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.UpdateTypeVersion Method

Updates the object and make this object use the latest Type definition for its class. The object now will now adjust for data attribute changes.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void UpdateTypeVersion ()
```

VB

```
Public Sub UpdateTypeVersion
```

C++

```
public:  
void UpdateTypeVersion ()
```

F#

```
member UpdateTypeVersion : unit -> unit
```

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.Write Method

By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page will also be written.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Write()
```

VB

```
Public Sub Write
```

C++

```
public:  
void Write()
```

F#

```
member Write : unit -> unit
```

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.WriteMe Method

Used by code generator.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual byte[] WriteMe(
    TypeVersion typeVersion,
    bool addShapeNumber,
    PageInfo pageInfo,
    IOptimizedPersistable owner,
    SessionBase session,
    bool inFlush
)
```

VB

```
Public Overridable Function WriteMe (
    typeVersion As TypeVersion,
    addShapeNumber As Boolean,
    pageInfo As PageInfo,
    owner As IOptimizedPersistable,
    session As SessionBase,
    inFlush As Boolean
) As Byte ()
```

C++

```
public:
virtual array<unsigned char>^ WriteMe(
    TypeVersion^ typeVersion,
    bool addShapeNumber,
    PageInfo^ pageInfo,
    IOptimizedPersistable^ owner,
    SessionBase^ session,
    bool inFlush
)
```

F#

```
abstract WriteMe :
    typeVersion : TypeVersion *
    addShapeNumber : bool *
    pageInfo : PageInfo *
    owner : IOptimizedPersistable *
    session : SessionBase *
    inFlush : bool -> byte[]
override WriteMe :
    typeVersion : TypeVersion *
    addShapeNumber : bool *
    pageInfo : PageInfo *
```

```
owner : IOptimizedPersistable *  
session : SessionBase *  
inFlush : bool -> byte[]
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

addShapeNumber

Type: [System.Boolean](#)

[Missing <param name="addShapeNumber"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

pageInfo

Type: [VelocityDb.PageInfo](#)

[Missing <param name="pageInfo"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

owner

Type: [VelocityDb.IOptimizedPersistable](#)

[Missing <param name="owner"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

inFlush

Type: [System.Boolean](#)

[Missing <param name="inFlush"/> documentation for "M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [Byte\[\]](#)

[Missing <returns> documentation for

"M:VelocityDb.OptimizedPersistable.WriteMe(VelocityDb.TypeInfo.TypeVersion,System.Boolean,VelocityDb.PageInfo,VelocityDb.IOptimizedPersistable,VelocityDb.Session.SessionBase,System.Boolean)"

]

Implements

[IOptimizedPersistable.WriteMe\(TypeVersion, Boolean, PageInfo, IOptimizedPersistable, SessionBase, Boolean\)](#)

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

OptimizedPersistable.WriteMeUsingSchemaReflection Method

Used by code generator. This is the default way of writing objects using schema information and .NET reflection info.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static byte[] WriteMeUsingSchemaReflection(
    TypeVersion typeVersion,
    IOptimizedPersistable pObj,
    bool addShapeNumber,
    PageInfo pageInfo,
    IOptimizedPersistable owner,
    SessionBase session,
    bool inFlush
)
```

VB

```
Public Shared Function WriteMeUsingSchemaReflection (
    typeVersion As TypeVersion,
    pObj As IOptimizedPersistable,
    addShapeNumber As Boolean,
    pageInfo As PageInfo,
    owner As IOptimizedPersistable,
    session As SessionBase,
    inFlush As Boolean
) As Byte()
```

C++

```
public:
static array<unsigned char>^ WriteMeUsingSchemaReflection(
    TypeVersion^ typeVersion,
    IOptimizedPersistable^ pObj,
    bool addShapeNumber,
    PageInfo^ pageInfo,
    IOptimizedPersistable^ owner,
    SessionBase^ session,
    bool inFlush
)
```

F#

```
static member WriteMeUsingSchemaReflection :
    typeVersion : TypeVersion *
    pObj : IOptimizedPersistable *
    addShapeNumber : bool *
    pageInfo : PageInfo *
    owner : IOptimizedPersistable *
    session : SessionBase *
```

```
inFlush : bool -> byte[]
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object being written

addShapeNumber

Type: [System.Boolean](#)

Shall we include bytes containing the type version of the object?

pageInfo

Type: [VelocityDb.PageInfo](#)

PageInfo for the current page

owner

Type: [VelocityDb.IOptimizedPersistable](#)

Owner of object being written

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

inFlush

Type: [System.Boolean](#)

Is page flushing OK while preparing this object to be written?

Return Value

Type: [Byte\[\]](#)

Bytes of this object

See Also

[OptimizedPersistable Class](#)

[VelocityDb Namespace](#)

Page Class

Each [Database](#) consist of a number of variable sized pages. A page can be compressed and may be encrypted and contains one or more objects. Page compression is controlled by the [DatabaseLocation](#) of the [Database](#) containing a page. Each page has a [PageInfo](#) that provides info about a page and can be updated to turn on/off compression and encryption for a specific page.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Page : IComparable<Page>, IEnumerable<IOptimizedPersistable>,
    IEnumerable, IDisposable
```

VB

```
Public Class Page
    Implements IComparable(Of Page), IEnumerable(Of IOptimizedPersistable),
    IEnumerable, IDisposable
```

C++


```
public ref class Page : IComparable<Page^>,
    IEnumerable<IOptimizedPersistable^>, IEnumerable, IDisposable
```

F#











```
type Page =
    class
        interface IComparable<Page>
        interface IEnumerable<IOptimizedPersistable>
        interface IEnumerable
        interface IDisposable
    end
```

The **Page** type exposes the following members.













Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|--------------------------------------------------------|
|  | Page | Internal use and usage in VelocityDbExteanions project |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------|
|  | ByteCount | |
|  | Database | The Database of this Page. |
|  | Id | The Oid (encoded as UInt64) of this page |
|  | IsUpdated | Is this Page updated? |
|  | Offset | Current page version starts at this offset in containing Database . |
|  | Oid | The Oid of this page |
|  | OnlyStub | Is this Page fully read into memory or just the PageInfo? |
|  | PageInfo | Gets the PageInfo for this Page |
|  | PageNumber | The page number of this Page in the containing Database |
|  | ShortId | The OidShort (encoded as UInt32) of this page |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|  | ClearCachedObjects | Clears all objects cached on the page but does not clear objects cached by weak references. |
|  | CompareTo | Compares Pages by PageNumber |
|  | DeleteObject | Use this as an alternative to Unpersist(SessionBase) or when using the interface class IOptimizedPersistable |
|  | Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |
|  | FinishUpCsvImport | Schema page requires special handling in Import from CSV files |
|  | GetEnumerator | Enumerates all objects on this Page |
|  | ObjectsLazyLoaded | Enumerates all objects on this Page. Object graph loaded up to specified max depth. |
|  | PageShortIds | Enumerates all objects short ids on this Page |
|  | Slot | Retrieves an object from persistent storage |
|  | StringToByteArray | Converts a string into an UTF8 encoded byte array |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |
|  | UnpersistObject | Use this as an alternative to Unpersist(SessionBase) or when using the interface class IOptimizedPersistable |

See Also

[VelocityDb Namespace](#)

Page Constructor

Internal use and usage in VelocityDbExtensions project

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Page(  
    Database db,  
    ushort pageNumber,  
    uint shapeNumber,  
    ushort objectsPerPage,  
    bool doCache = true  
)
```

VB

```
Public Sub New (  
    db As Database,  
    pageNumber As UShort,  
    shapeNumber As UInteger,  
    objectsPerPage As UShort,  
    Optional doCache As Boolean = true  
)
```

C++

```
public:  
Page(  
    Database^ db,  
    unsigned short pageNumber,  
    unsigned int shapeNumber,  
    unsigned short objectsPerPage,  
    bool doCache = true  
)
```

F#

```
new :  
    db : Database *  
    pageNumber : uint16 *  
    shapeNumber : uint32 *  
    objectsPerPage : uint16 *  
    ?doCache : bool  
(* Defaults:  
    let _doCache = defaultArg doCache true  
*)  
-> Page
```

Parameters

db

Type: [VelocityDb.Database](#)
[Database](#)

this [Page](#) belongs to

pageNumber

Type: [System.UInt16](#)

Page number of this page

shapeNumber

Type: [System.UInt32](#)

Short Id of object schema type if page contains just a single type of objects

objectsPerPage

Type: [System.UInt16](#)

Max how many objects to store on this page

doCache (Optional)

Type: [System.Boolean](#)

Cache this page in its database?

See Also











[Page Class](#)

[VelocityDb Namespace](#)

Page.Page Properties

The [Page](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------|
|  | ByteCount | |
|  | Database | The Database of this Page. |
|  | Id | The Oid (encoded as UInt64) of this page |
|  | IsUpdated | Is this Page updated? |
|  | Offset | Current page version starts at this offset in conatining Database . |
|  | Oid | The Oid of this page |
|  | OnlyStub | Is this Page fully read into memory or just the PageInfo? |
|  | PageInfo | Gets the PageInfo for this Page |
|  | PageNumber | The page number of this Page in the containing Database |
|  | ShortId | The OidShort (encoded as UInt32) of this page |

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.ByteCount Property

[Missing <summary> documentation for "P:VelocityDb.Page.ByteCount"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ByteCount { get; }
```

VB

```
Public ReadOnly Property ByteCount As UInteger  
    Get
```

C++

```
public:  
property unsigned int ByteCount {  
    unsigned int get ();  
}
```

F#

```
member ByteCount : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Database Property

The **Database** of this Page.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Database Database { get; }
```

VB

```
Public ReadOnly Property Database As Database  
    Get
```

C++

```
public:  
property Database^ Database {  
    Database^ get ();  
}
```

F#

```
member Database : Database with get
```

Property Value

Type: [Database](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Id Property

The Oid (encoded as UInt64) of this page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Id { get; }
```

VB

```
Public ReadOnly Property Id As ULong  
    Get
```

C++

```
public:  
property unsigned long long Id {  
    unsigned long long get ();  
}
```

F#

```
member Id : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.IsUpdated Property

Is this Page updated?

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsUpdated { get; }
```

VB

```
Public ReadOnly Property IsUpdated As Boolean  
    Get
```

C++

```
public:  
property bool IsUpdated {  
    bool get ();  
}
```

F#

```
member IsUpdated : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Offset Property

Current page version starts at this offset in conatining [Database](#).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Offset { get; }
```

VB

```
Public ReadOnly Property Offset As Long  
    Get
```

C++

```
public:  
property long long Offset {  
    long long get ();  
}
```

F#

```
member Offset : int64 with get
```

Property Value

Type: [Int64](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Oid Property

The Oid of this page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid Oid { get; }
```

VB

```
Public ReadOnly Property Oid As Oid  
    Get
```

C++

```
public:  
property Oid Oid {  
    Oid get ();  
}
```

F#

```
member Oid : Oid with get
```

Property Value

Type: [Oid](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.OnlyStub Property

Is this Page fully read into memory or just the PageInfo?

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool OnlyStub { get; }
```

VB

```
Public ReadOnly Property OnlyStub As Boolean  
    Get
```

C++

```
public:  
property bool OnlyStub {  
    bool get ();  
}
```

F#

```
member OnlyStub : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.PageInfo Property

Gets the PageInfo for this Page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageInfo PageInfo { get; }
```

VB

```
Public ReadOnly Property PageInfo As PageInfo  
    Get
```

C++

```
public:  
property PageInfo^ PageInfo {  
    PageInfo^ get ();  
}
```

F#

```
member PageInfo : PageInfo with get
```

Property Value

Type: [PageInfo](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.PageNumber Property

The page number of this Page in the containing [Database](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort PageNumber { get; }
```

VB

```
Public ReadOnly Property PageNumber As UShort  
    Get
```

C++

```
public:  
property unsigned short PageNumber {  
    unsigned short get ();  
}
```

F#

```
member PageNumber : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.ShortId Property

The OidShort (encoded as UInt32) of this page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ShortId { get; }
```

VB

```
Public ReadOnly Property ShortId As UInteger  
    Get
```

C++

```
public:  
property unsigned int ShortId {  
    unsigned int get ();  
}
```

F#

```
member ShortId : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also













[Page Class](#)

[VelocityDb Namespace](#)

Page.Page Methods

The [Page](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|  | ClearCachedObjects | Clears all objects cached on the page but does not clear objects cached by weak references. |
|  | CompareTo | Compares Pages by PageNumber |
|  | DeleteObject | Use this as an alternative to Unpersist(SessionBase) or when using the interface class IOptimizedPersistable |
|  | Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |
|  | FinishUpCsvImport | Schema page requires special handling in Import from CSV files |
|  | GetEnumerator | Enumerates all objects on this Page |
|  | ObjectsLazyLoaded | Enumerates all objects on this Page. Object graph loaded up to specified max depth. |
|  | PageShortIds | Enumerates all objects short ids on this Page |
|  | Slot | Retrieves an object from persistent storage |
|  | StringToByteArray | Converts a string into an UTF8 encoded byte array |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |
|  | UnpersistObject | Use this as an alternative to Unpersist(SessionBase) or when using the interface class IOptimizedPersistable |

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.ClearCachedObjects Method

Clears all objects cached on the page but does not clear objects cached by weak references.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort ClearCachedObjects(  
    SessionBase session  
)
```

VB

```
Public Function ClearCachedObjects (  
    session As SessionBase  
) As UShort
```

C++

```
public:  
unsigned short ClearCachedObjects(  
    SessionBase^ session  
)
```

F#

```
member ClearCachedObjects :  
    session : SessionBase -> uint16
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

Return Value

Type: [UInt16](#)

[Missing <returns> documentation for "M:VelocityDb.Page.ClearCachedObjects(VelocityDb.Session.SessionBase)"]

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.CompareTo Method

Compares Pages by PageNumber

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    Page otherPage  
)
```

VB

```
Public Function CompareTo (  
    otherPage As Page  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Page^ otherPage  
) sealed
```

F#

```
abstract CompareTo :  
    otherPage : Page -> int  
override CompareTo :  
    otherPage : Page -> int
```

Parameters

otherPage

Type: [VelocityDb.Page](#)

The Page to compare with

Return Value

Type: [Int32](#)

-1 if this PageNumber is less than obj.PageNumber, 0 if equal and 1 if greater than

Implements

[IComparable\(T\).CompareTo\(T\)](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.DeleteObject Method

Use this as an alternative to [Unpersist\(SessionBase\)](#) or when using the interface class [IOptimizedPersistable](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeleteObject (  
    ulong id  
)
```

VB

```
Public Sub DeleteObject (  
    id As ULong  
)
```

C++

```
public:  
void DeleteObject (  
    unsigned long long id  
)
```

F#

```
member DeleteObject :  
    id : uint64 -> unit
```

Parameters

id

Type: [System.UInt64](#)

The object Id of the object to be deleted

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.FinishUpCsvImport Method

Schema page requires special handling in Import from CSV files

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FinishUpCsvImport ()
```

VB

```
Public Sub FinishUpCsvImport
```

C++

```
public:  
void FinishUpCsvImport ()
```

F#

```
member FinishUpCsvImport : unit -> unit
```

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.GetEnumerator Method

Enumerates all objects on this Page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<IOptimizedPersistable> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of IOptimizedPersistable)
```

C++

```
public:  
virtual IEnumerator<IOptimizedPersistable^> GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<IOptimizedPersistable>  
override GetEnumerator : unit -> IEnumerator<IOptimizedPersistable>
```

Return Value

Type: [IEnumerator<IOptimizedPersistable>](#)

Enumeration of all slots on this Page

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.ObjectsLazyLoaded Method

Enumerates all objects on this Page. Object graph loaded up to specified max depth.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IOptimizedPersistable> ObjectsLazyLoaded(  
    int depthToLoad = 0  
)
```

VB

```
Public Function ObjectsLazyLoaded (  
    Optional depthToLoad As Integer = 0  
) As IEnumerable(Of IOptimizedPersistable)
```

C++

```
public:  
IEnumerable<IOptimizedPersistable^>^ ObjectsLazyLoaded(  
    int depthToLoad = 0  
)
```

F#

```
member ObjectsLazyLoaded :  
    ?depthToLoad : int  
(* Defaults:  
    let _depthToLoad = defaultArg depthToLoad 0  
*)  
-> IEnumerable<IOptimizedPersistable>
```

Parameters

depthToLoad (Optional)

Type: [System.Int32](#)

Max how many levels deep to load each object

Return Value

Type: [IEnumerable\(IOptimizedPersistable\)](#)

Enumeration of all slots on this Page

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.PageShortIds Method

Enumerates all objects short ids on this Page

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<uint> PageShortIds ()
```

VB

```
Public Function PageShortIds As IEnumerable (Of UInteger)
```

C++

```
public:  
IEnumerable<unsigned int>^ PageShortIds ()
```

F#

```
member PageShortIds : unit -> IEnumerable<uint32>
```

Return Value

Type: [IEnumerable\(UInt32\)](#)

Enumeration of all slot short ids on this Page

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.Slot Method

Retrieves an object from persistent storage

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IOptimizedPersistable Slot(
    ulong oid,
    bool openRefs,
    SessionBase session,
    Schema schema,
    bool iteration,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad = 2147483647,
    TypeVersion shape = null
)
```

VB

```
Public Function Slot (
    oid As ULong,
    openRefs As Boolean,
    session As SessionBase,
    schema As Schema,
    iteration As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    Optional graphDepthToLoad As Integer = 2147483647,
    Optional shape As TypeVersion = Nothing
) As IOptimizedPersistable
```

C++

```
public:
IOptimizedPersistable^ Slot(
    unsigned long long oid,
    bool openRefs,
    SessionBase^ session,
    Schema^ schema,
    bool iteration,
    List<IOptimizedPersistable^>^ toLoadMembers,
    int graphDepth,
    int graphDepthToLoad = 2147483647,
    TypeVersion^ shape = nullptr
)
```

F#

```
member Slot :
    oid : uint64 *
```

VelocityDB Class Library

```
    openRefs : bool *
    session : SessionBase *
    schema : Schema *
    iteration : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    graphDepth : int *
    ?graphDepthToLoad : int *
    ?shape : TypeVersion
(* Defaults:
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
    let_shape = defaultArg shape null
*)
-> IOptimizedPersistable
```

Parameters

oid

Type: [System.UInt64](#)

Id of object to retrieve

openRefs

Type: [System.Boolean](#)

Shall we open references from this object?

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active schema

iteration

Type: [System.Boolean](#)

Is this called as part of an iteration

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

List of members to load

graphDepth

Type: [System.Int32](#)

The current load graph depth

graphDepthToLoad (Optional)

Type: [System.Int32](#)

Max graph depth to load

shape (Optional)

Type: [VelocityDb.TypeInfo.TypeVersion](#)

VelocityDB Class Library

The type version of the object to retrieve

Return Value

Type: [IOptimizedPersistable](#)

The retrieved object

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.StringToByteArray Method

Converts a string into an UTF8 encoded byte array

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static byte[] StringToByteArray(  
    string str  
)
```

VB

```
Public Shared Function StringToByteArray (  
    str As String  
) As Byte()
```

C++

```
public:  
static array<unsigned char>^ StringToByteArray(  
    String^ str  
)
```

F#

```
static member StringToByteArray :  
    str : string -> byte[]
```

Parameters

str

Type: [System.String](#)

a string to convert

Return Value

Type: [Byte\[\]](#)

a UTF8 encoding

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[Page Class](#)

[VelocityDb Namespace](#)

Page.UnpersistObject Method

Use this as an alternative to [Unpersist\(SessionBase\)](#) or when using the interface class [IOptimizedPersistable](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void UnpersistObject (
    IOptimizedPersistable pObj
)
```

VB

```
Public Sub UnpersistObject (
    pObj As IOptimizedPersistable
)
```

C++

```
public:
void UnpersistObject (
    IOptimizedPersistable^ pObj
)
```

F#

```
member UnpersistObject :
    pObj : IOptimizedPersistable -> unit
```

Parameters

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to be unpersisted

See Also

[Page Class](#)

[VelocityDb Namespace](#)

PageCache Class

Maintains string references to [Pages](#) within a [Database](#)

Inheritance Hierarchy

[System.Object](#)

VelocityDb.PageCache

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PageCache : IEnumerable<Page>,
    IEnumerable
```

VB

```
Public Class PageCache
    Implements IEnumerable(Of Page), IEnumerable
```

C++



```
public ref class PageCache : IEnumerable<Page^>,
    IEnumerable
```

F#


```
type PageCache =
    class
        interface IEnumerable<Page>
        interface IEnumerable
    end
```

The **PageCache** type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-------------------------------------|
|  | Item | Get or set a page in the page cache |
|  | Length | Max number of pages in page cache |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------|
|  | GetEnumerator | Enumeration of all objects of the cached Pages |

VelocityDB Class Library



See Also

[VelocityDb Namespace](#)

PageCache.PageCache Properties

The [PageCache](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------------------------------|
|  | Item | Get or set a page in the page cache |
|  | Length | Max number of pages in page cache |

See Also

[PageCache Class](#)

[VelocityDb Namespace](#)

PageCache.Item Property

Get or set a page in the page cache

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Page this[
    ushort pageNumber
] { get; set; }
```

VB

```
Public Default Property Item (
    pageNumber As UShort
) As Page
    Get
    Set
```

C++

```
public:
property Page^ default[unsigned short pageNumber] {
    Page^ get (unsigned short pageNumber);
    void set (unsigned short pageNumber, Page^ value);
}
```

F#

```
member Item : Page with get, set
```

Parameters

pageNumber

Type: [System.UInt16](#)

Page number within a database for requested page

Return Value

Type: [Page](#)

A page if it exist in the page cache

See Also

[PageCache Class](#)

[VelocityDb Namespace](#)

PageCache.Length Property

Max number of pages in page cache

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Length { get; }
```

VB

```
Public ReadOnly Property Length As Integer  
    Get
```

C++

```
public:  
property int Length {  
    int get ();  
}
```

F#

```
member Length : int with get
```

Property Value

Type: [Int32](#)

See Also


[PageCache Class](#)

[VelocityDb Namespace](#)

PageCache.PageCache Methods

The [PageCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------|
|  | GetEnumerator | Enumeration of all objects of the cached Pages |

See Also

[PageCache Class](#)

[VelocityDb Namespace](#)

PageCache.GetEnumerator Method

Enumeration of all objects of the cached [Pages](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator
```

C++

```
public:  
virtual IEnumerator^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator  
override GetEnumerator : unit -> IEnumerator
```

Return Value

Type: [IEnumerator](#)

The enumeration

Implements

[IEnumerable.GetEnumerator\(\)](#)

See Also

[PageCache Class](#)

[VelocityDb Namespace](#)

PageInfo Class

Each [Page](#) has a PageInfo that provides info about a page and can be used to turn on/off compression and encryption for a page.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.PageInfo

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class PageInfo
```










```
VB
Public Class PageInfo
```

```
C++
public ref class PageInfo
```



```
F#
type PageInfo = class end
```

The **PageInfo** type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  Compressed | A Page can be compressed or not. |
|  Encryption | A Page can be encrypted with DES or not. |
|  FirstFreeSlot | The first available page on the Page . |
|  NoFreeSlotInBetweenInUseSlots | Is there no gaps of unused slots in between the in use slots (all contiguous in use slots)? |
|  NumberOfSlots | The number of objects/slots on the Page . |
|  OnDiskSize | The byte size of the Page when it was last written to disk. |
|  ShapeNumber | If non 0, all objects on the page have the same Type. |
|  UncompressedSize | The byte size of the Page when it was last written to disk (before any compression was applied). |
|  VersionNumber | Each Page gets its page version incremented in every transaction where the it is updated. |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|-----------------------------------------------------------------------------|
|  | s_pageInfoByteCount | The size of the PageInfo at the start of each Page . |
|  | s_pageInfoByteCountV1 | |










See Also

[VelocityDb Namespace](#)

PageInfo.PageInfo Properties

The [PageInfo](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | Compressed | A Page can be compressed or not. |
|  | Encryption | A Page can be encrypted with DES or not. |
|  | FirstFreeSlot | The first available page on the Page . |
|  | NoFreeSlotInBetweenInUseSlots | Is there no gaps of unused slots in between the in use slots (all contiguous in use slots)? |
|  | NumberOfSlots | The number of objects/slots on the Page . |
|  | OnDiskSize | The byte size of the Page when it was last written to disk. |
|  | ShapeNumber | If non 0, all objects on the page have the same Type. |
|  | UncompressedSize | The byte size of the Page when it was last written to disk (before any compression was applied). |
|  | VersionNumber | Each Page gets its page version incremented in every transaction where the it is updated. |

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.Compressed Property

A [Page](#) can be compressed or not.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageInfo.compressionKind Compressed { get; set; }
```

VB

```
Public Property Compressed As PageInfo.compressionKind  
    Get  
    Set
```

C++

```
public:  
property PageInfo.compressionKind Compressed {  
    PageInfo.compressionKind get ();  
    void set (PageInfo.compressionKind value);  
}
```

F#

```
member Compressed : PageInfo.compressionKind with get, set
```

Property Value

Type: [PageInfo.compressionKind](#)

The [Boolean](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.Encryption Property

A [Page](#) can be encrypted with DES or not.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageInfo.encryptionKind Encryption { get; set; }
```

VB

```
Public Property Encryption As PageInfo.encryptionKind  
    Get  
    Set
```

C++

```
public:  
property PageInfo.encryptionKind Encryption {  
    PageInfo.encryptionKind get ();  
    void set (PageInfo.encryptionKind value);  
}
```

F#

```
member Encryption : PageInfo.encryptionKind with get, set
```

Property Value

Type: [PageInfo.encryptionKind](#)

The [PageInfo.encryptionKind](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.FirstFreeSlot Property

The first available page on the [Page](#).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort FirstFreeSlot { get; set; }
```

VB

```
Public Property FirstFreeSlot As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short FirstFreeSlot {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member FirstFreeSlot : uint16 with get, set
```

Property Value

Type: [UInt16](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.NoFreeSlotInBetweenInUseSlots Property

Is there no gaps of unused slots in between the in use slots (all contiguous in use slots)?

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool NoFreeSlotInBetweenInUseSlots { get; }
```

VB

```
Public ReadOnly Property NoFreeSlotInBetweenInUseSlots As Boolean  
    Get
```

C++

```
public:  
property bool NoFreeSlotInBetweenInUseSlots {  
    bool get ();  
}
```

F#

```
member NoFreeSlotInBetweenInUseSlots : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.NumberOfSlots Property

The number of objects/slots on the [Page](#).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort NumberOfSlots { get; set; }
```

VB

```
Public Property NumberOfSlots As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short NumberOfSlots {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member NumberOfSlots : uint16 with get, set
```

Property Value

Type: [UInt16](#)

The [UInt16](#) of objects/slots on the page

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.OnDiskSize Property

The byte size of the [Page](#) when it was last written to disk.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint OnDiskSize { get; }
```

VB

```
Public ReadOnly Property OnDiskSize As UInteger  
    Get
```

C++

```
public:  
property unsigned int OnDiskSize {  
    unsigned int get ();  
}
```

F#

```
member OnDiskSize : uint32 with get
```

Property Value

Type: [UInt32](#)

The [UInt32](#) of bytes

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.ShapeNumber Property

If non 0, all objects on the page have the same Type.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ShapeNumber { get; }
```

VB

```
Public ReadOnly Property ShapeNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int ShapeNumber {  
    unsigned int get ();  
}
```

F#

```
member ShapeNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.UncompressedSize Property

The byte size of the [Page](#) when it was last written to disk (before any compression was applied).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint UncompressedSize { get; }
```

VB

```
Public ReadOnly Property UncompressedSize As UInteger  
    Get
```

C++

```
public:  
property unsigned int UncompressedSize {  
    unsigned int get ();  
}
```

F#

```
member UncompressedSize : uint32 with get
```

Property Value

Type: [UInt32](#)

The [UInt32](#) of bytes

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.VersionNumber Property

Each [Page](#) gets its page version incremented in every transaction where the it is updated.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong VersionNumber { get; set; }
```

VB

```
Public Property VersionNumber As ULong  
    Get  
    Set
```

C++

```
public:  
property unsigned long long VersionNumber {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
member VersionNumber : uint64 with get, set
```

Property Value

Type: [UInt64](#)

The current [UInt64](#) page version

See Also





[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.PageInfo Fields

The [PageInfo](#) type exposes the following members.

Fields

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------|
|   | s_pageInfoByteCount | The size of the PageInfo at the start of each Page . |
|   | s_pageInfoByteCountV1 | |

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.s_pageInfoByteCount Field

The size of the [PageInfo](#) at the start of each [Page](#).

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_pageInfoByteCount
```

VB

```
Public Shared ReadOnly s_pageInfoByteCount As UShort
```

C++

```
public:  
static initonly unsigned short s_pageInfoByteCount
```

F#

```
static val s_pageInfoByteCount: uint16
```

Field Value

Type: [UInt16](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.s_pageInfoByteCountV1 Field

[Missing <summary> documentation for "F:VelocityDb.PageInfo.s_pageInfoByteCountV1"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_pageInfoByteCountV1
```

VB

```
Public Shared ReadOnly s_pageInfoByteCountV1 As UShort
```

C++

```
public:  
static initonly unsigned short s_pageInfoByteCountV1
```

F#

```
static val s_pageInfoByteCountV1: uint16
```

Field Value

Type: [UInt16](#)

See Also

[PageInfo Class](#)

[VelocityDb Namespace](#)

PageInfo.compressionKind Enumeration

Selection of choices for [Page](#) compression.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum compressionKind
```

VB

```
Public Enumeration compressionKind
```

C++

```
public enum class compressionKind
```

F#

```
type compressionKind
```

Members

| Member name | Value | Description |
|-------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| None | 0 | No Page compression |
| GZip | 1 | GZipStreamPage compression |
| LZ4 | 2 | LZ4 (safe) Page compression LZ4 - Fast LZ compression algorithm Copyright (C) 2011-2012, Yann Collet. BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php) Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF |

| | | |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. You can contact the author at : - LZ4 homepage : http://fastcompression.blogspot.com/p/lz4.html - LZ4 source repository : http://code.google.com/p/lz4/ Port to .NET https://lz4net.codeplex.com/ Copyright (c) 2013, Milosz Krajewski |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

See Also

[VelocityDb Namespace](#)

PageInfo.encryptedKind Enumeration

Use **desEncrypted** for encrypted pages, set encryption private key in [DatabaseLocation](#). Use **noEncryption** when no encryption is desired. Other choices in this enum are not yet publicly available. Other kinds of encryption can be provided upon request. We can even make the interface public allowing application defined encryption.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum encryptionKind
```

VB

```
Public Enumeration encryptionKind
```

C++

```
public enum class encryptionKind
```

F#

```
type encryptionKind
```

Members

| Member name | Value | Description |
|-------------------------|-------|------------------------------------------------------------------------------------|
| noEncryption | 0 | No Page encryption |
| desEncrypted | 1 | Use DES Page encryption |
| rsaEncrypted2048 | 2 | Use RSA 2048bits Page encryption (not yet made publicly available) |
| rsaSigned2048 | 3 | Use RSA 2048bits Page signing (not yet made publicly available) |

See Also

[VelocityDb Namespace](#)

PageOffset Class

Internally Used within VelocityDB and its extensions

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.PageOffset

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PageOffset : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class PageOffset  
    Inherits OptimizedPersistable
```

C++



```
[SerializableAttribute]  
public ref class PageOffset : public OptimizedPersistable
```

F#

```
[<SerializableAttribute>]  
type PageOffset =  
    class  
        inherit OptimizedPersistable  
    end
```




The **PageOffset** type exposes the following members.

Constructors





| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------------------------|
|  | PageOffset(UInt16) | Initializes a new instance of the PageOffset class |
|  | PageOffset(SessionBase) | Initializes a new instance of the PageOffset class |

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|-------------|
|  | Count | |

| | | |
|-----------------------------------------------------------------------------------|-----------------------------------|--|
|  | HighestPageNumber | |
|  | InOffsetOrder | |
|  | Item | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add | |
|  | OffsetLookup | |
|  | ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  | Remove | |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb Namespace](#)

PageOffset Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------|
|  | PageOffset(UInt16) | Initializes a new instance of the PageOffset class |
|  | PageOffset(SessionBase) | Initializes a new instance of the PageOffset class |

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset Constructor (UInt16)

Initializes a new instance of the [PageOffset](#) class

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageOffset(  
    ushort preSize = 1000  
)
```

VB

```
Public Sub New (  
    Optional preSize As UShort = 1000  
)
```

C++

```
public:  
PageOffset(  
    unsigned short preSize = 1000  
)
```

F#

```
new :  
    ?preSize : uint16  
(* Defaults:  
    let_preSize = defaultArg preSize 1000  
)  
-> PageOffset
```

Parameters

preSize (Optional)

Type: [System.UInt16](#)

[Missing <param name="preSize"/> documentation for "M:VelocityDb.PageOffset.#ctor(System.UInt16)"]

See Also

[PageOffset Class](#)

[PageOffset Overload](#)

[VelocityDb Namespace](#)

PageOffset Constructor (SessionBase)

Initializes a new instance of the [PageOffset](#) class

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageOffset (  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    session As SessionBase  
)
```

C++

```
public:  
PageOffset (  
    SessionBase^ session  
)
```

F#

```
new :  
    session : SessionBase -> PageOffset
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.PageOffset.#ctor(VelocityDb.Session.SessionBase)"]

See Also

[PageOffset Class](#)





[PageOffset Overload](#)

[VelocityDb Namespace](#)

PageOffset.PageOffset Properties

The [PageOffset](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | Count | |
|  | HighestPageNumber | |
|  | InOffsetOrder | |
|  | Item | |

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.Count Property

[Missing <summary> documentation for "P:VelocityDb.PageOffset.Count"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
property int Count {  
    int get ();  
}
```

F#

```
member Count : int with get
```

Property Value

Type: [Int32](#)

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.HighestPageNumber Property

[Missing <summary> documentation for "P:VelocityDb.PageOffset.HighestPageNumber"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort HighestPageNumber { get; }
```

VB

```
Public ReadOnly Property HighestPageNumber As UShort  
    Get
```

C++

```
public:  
property unsigned short HighestPageNumber {  
    unsigned short get ();  
}
```

F#

```
member HighestPageNumber : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.InOffsetOrder Property

[Missing <summary> documentation for "P:VelocityDb.PageOffset.InOffsetOrder"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMap<long, ushort> InOffsetOrder { get; }
```

VB

```
Public ReadOnly Property InOffsetOrder As SortedMap(Of Long, UShort)  
    Get
```

C++

```
public:  
property SortedMap<long long, unsigned short>^ InOffsetOrder {  
    SortedMap<long long, unsigned short>^ get ();  
}
```

F#

```
member InOffsetOrder : SortedMap<int64, uint16> with get
```

Property Value

Type: [SortedMap\(Int64, UInt16\)](#)

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.Item Property

[Missing <summary> documentation for "P:VelocityDb.PageOffset.Item(System.UInt16)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long this[
    ushort pageNumber
] { get; set; }
```

VB

```
Public Default Property Item (
    pageNumber As UShort
) As Long
    Get
    Set
```

C++

```
public:
property long long default[unsigned short pageNumber] {
    long long get (unsigned short pageNumber);
    void set (unsigned short pageNumber, long long value);
}
```

F#

```
member Item : int64 with get, set
```

Parameters

pageNumber

Type: [System.UInt16](#)

Property Value

Type: [Int64](#)

See Also





[PageOffset Class](#)

[VelocityDb Namespace](#)



PageOffset.PageOffset Methods

The [PageOffset](#) type exposes the following members.

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | |
|  OffsetLookup | |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  Remove | |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.Add Method

[Missing <summary> documentation for "M:VelocityDb.PageOffset.Add(System.UInt16,System.Int64)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Add(  
    ushort pageNumber,  
    long pageOffset  
)
```

VB

```
Public Sub Add (  
    pageNumber As UShort,  
    pageOffset As Long  
)
```

C++

```
public:  
void Add(  
    unsigned short pageNumber,  
    long long pageOffset  
)
```

F#

```
member Add :  
    pageNumber : uint16 *  
    pageOffset : int64 -> unit
```

Parameters

pageNumber

Type: [System.UInt16](#)

[Missing <param name="pageNumber"/> documentation for "M:VelocityDb.PageOffset.Add(System.UInt16,System.Int64)"]

pageOffset

Type: [System.Int64](#)

[Missing <param name="pageOffset"/> documentation for "M:VelocityDb.PageOffset.Add(System.UInt16,System.Int64)"]

See Also

[PageOffset Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

PageOffset.OffsetLookup Method

[Missing <summary> documentation for "M:VelocityDb.PageOffset.OffsetLookup(System.UInt16)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long OffsetLookup(  
    ushort pageNumber  
)
```

VB

```
Public Function OffsetLookup (  
    pageNumber As UShort  
) As Long
```

C++

```
public:  
long long OffsetLookup(  
    unsigned short pageNumber  
)
```

F#

```
member OffsetLookup :  
    pageNumber : uint16 -> int64
```

Parameters

pageNumber

Type: [System.UInt16](#)

[Missing <param name="pageNumber"/> documentation for "M:VelocityDb.PageOffset.OffsetLookup(System.UInt16)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityDb.PageOffset.OffsetLookup(System.UInt16)"]

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```

    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
) override

```

F#

```

abstract ReadMe :
    typeVersion : TypeVersion *
    memberBytes : byte[] *
    offset : int byref *
    session : SessionBase *
    page : Page *
    useOidShort : bool *
    schema : Schema *
    openRefs : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    graphDepth : int *
    graphDepthToLoad : int *
    primitivesOnly : bool -> unit
override ReadMe :
    typeVersion : TypeVersion *
    memberBytes : byte[] *
    offset : int byref *
    session : SessionBase *
    page : Page *
    useOidShort : bool *
    schema : Schema *
    openRefs : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    graphDepth : int *
    graphDepthToLoad : int *
    primitivesOnly : bool -> unit

```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,V

elocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.PageOffset.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

PageOffset.Remove Method

[Missing <summary> documentation for "M:VelocityDb.PageOffset.Remove(System.UInt16)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Remove(  
    ushort pageNumber  
)
```

VB

```
Public Function Remove (  
    pageNumber As UShort  
) As Boolean
```

C++

```
public:  
bool Remove(  
    unsigned short pageNumber  
)
```

F#

```
member Remove :  
    pageNumber : uint16 -> bool
```

Parameters

pageNumber

Type: [System.UInt16](#)

[Missing <param name="pageNumber"/> documentation for "M:VelocityDb.PageOffset.Remove(System.UInt16)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.PageOffset.Remove(System.UInt16)"]

See Also

[PageOffset Class](#)

[VelocityDb Namespace](#)

Placement Class

This class is used when deciding where to place a new object persistently. That is we have a choice of database number, page number and page number. There is other ways to control the placement including [ObjectsPerPage](#), [PagesPerDatabase](#), and it is also possible to override [Persist\(Placement, SessionBase, Boolean, Boolean, Queue\(IOptimizedPersistable\)\)](#).

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Placement

[VelocityDb.AutoPlacement](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class Placement : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class Placement
    Inherits OptimizedPersistable
```

C++



```
[SerializableAttribute]
public ref class Placement : public OptimizedPersistable
```

F#










```
[<SerializableAttribute>]
type Placement =
    class
        inherit OptimizedPersistable
    end
```

The **Placement** type exposes the following members.





Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|  | Placement(SessionBase, IOptimizedPersistable, IOptimizedPersistable, Boolean, UInt32, Boolean) | Creates a Placement used for choosing a place to persist objects |
|  | Placement(UInt32, UInt16, UInt16, UInt16, UInt16, Boolean, Boolean, UInt32, Boolean, Boolean) | Creates a Placement used for choosing a place to persist objects |



Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Gets/sets permission to create multiple types on the same page. |
|  EndDatabaseNumber | The last Database number acceptable for placement |
|  MaxObjectsPerPage | Gets/Sets the limit on how many objects to create on a single page. |
|  MaxPagesPerDatabase | Gets/sets the limit on the number of pages per database. |
|  StartDatabaseNumber | The first/original Database number used by this Placement |
|  StartPageNumber | The first/original Page number used by this Placement |
|  TryDatabaseNumber | Gets the current Database number to try with first for object placement. |
|  TryPageNumber | Gets/sets the page number to try with first when persisting a new object. |
|  TrySlotNumber | Gets/sets the page number to try next when persisting an object. |



Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IncrementTryDatabaseNumber | Incremnts the Database number used when placing (persisting) objects |
|  IncrementTryPageNumber | Increments the tryPageNumber by 1 |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |

Fields

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
|  DefaultPlacementDatabaseNumber | If no other Database number is selected when persisting an object then this number is choosen. |
|  DefaultPlacementPage | If no other Page number is selected when persisting an object then this number is choosen. |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb Namespace](#)

Placement Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|  | Placement(SessionBase, IOptimizedPersistable, IOptimizedPersistable, Boolean, UInt32, Boolean) | Creates a Placement used for choosing a place to persist objects |
|  | Placement(UInt32, UInt16, UInt16, UInt16, UInt16, Boolean, Boolean, UInt32, Boolean, Boolean) | Creates a Placement used for choosing a place to persist objects |

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement Constructor (SessionBase, IOptimizedPersistable, IOptimizedPersistable, Boolean, UInt32, Boolean)

Creates a Placement used for choosing a place to persist objects

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Placement (
    SessionBase session,
    IOptimizedPersistable placementProviderObject,
    IOptimizedPersistable objectToPlace,
    bool persistRefs = false,
    uint maxNumberOfDatabases = 4294967295,
    bool flushFullPages = true
)
```

VB

```
Public Sub New (
    session As SessionBase,
    placementProviderObject As IOptimizedPersistable,
    objectToPlace As IOptimizedPersistable,
    Optional persistRefs As Boolean = false,
    Optional maxNumberOfDatabases As UIntInteger = 4294967295,
    Optional flushFullPages As Boolean = true
)
```

C++

```
public:
Placement (
    SessionBase^ session,
    IOptimizedPersistable^ placementProviderObject,
    IOptimizedPersistable^ objectToPlace,
    bool persistRefs = false,
    unsigned int maxNumberOfDatabases = 4294967295,
    bool flushFullPages = true
)
```

F#

```
new :
    session : SessionBase *
    placementProviderObject : IOptimizedPersistable *
    objectToPlace : IOptimizedPersistable *
    ?persistRefs : bool *
    ?maxNumberOfDatabases : uint32 *
    ?flushFullPages : bool
(* Defaults:
    let _persistRefs = defaultArg persistRefs false
```



```
        let_maxNumberOfDatabases = defaultArg maxNumberOfDatabases 4294967295
        let_flushFullPages = defaultArg flushFullPages true
    *)
-> Placement
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session being used

placementProviderObject

Type: [VelocityDb.IOptimizedPersistable](#)

If persistent, placement uses this object's db, page and slot as a starting point for placements

objectToPlace

Type: [VelocityDb.IOptimizedPersistable](#)

An object of the type we are going to persist with this placement object

persistRefs (Optional)

Type: [System.Boolean](#)

Is the placement going to request that referenced objects also get persisted?

maxNumberOfDatabases (Optional)

Type: [System.UInt32](#)

Maximum number of databases to try for placement

flushFullPages (Optional)

Type: [System.Boolean](#)

Shall a detected full page be flushed if such page is found while persisting an object with this placement

See Also

[Placement Class](#)

[Placement Overload](#)

[VelocityDb Namespace](#)

Placement Constructor (UInt32, UInt16, UInt16, UInt16, UInt16, Boolean, Boolean, UInt32, Boolean, Boolean)

Creates a Placement used for choosing a place to persist objects

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Placement(  
    uint db,  
    ushort page = 1,  
    ushort slot = 1,  
    ushort objectsPerPage = 10000,  
    ushort pagesPerDatabase = 10000,  
    bool persistRefs = false,  
    bool tryOtherDatabaseIfLockConflict = true,  
    uint maxNumberOfDatabases = 4294967295,  
    bool allowOtherTypesOnSamePage = true,  
    bool flushFullPages = true  
)
```

VB

```
Public Sub New (  
    db As UInteger,  
    Optional page As UShort = 1,  
    Optional slot As UShort = 1,  
    Optional objectsPerPage As UShort = 10000,  
    Optional pagesPerDatabase As UShort = 10000,  
    Optional persistRefs As Boolean = false,  
    Optional tryOtherDatabaseIfLockConflict As Boolean = true,  
    Optional maxNumberOfDatabases As UInteger = 4294967295,  
    Optional allowOtherTypesOnSamePage As Boolean = true,  
    Optional flushFullPages As Boolean = true  
)
```

C++

```
public:  
Placement(  
    unsigned int db,  
    unsigned short page = 1,  
    unsigned short slot = 1,  
    unsigned short objectsPerPage = 10000,  
    unsigned short pagesPerDatabase = 10000,  
    bool persistRefs = false,  
    bool tryOtherDatabaseIfLockConflict = true,  
    unsigned int maxNumberOfDatabases = 4294967295,  
    bool allowOtherTypesOnSamePage = true,  
    bool flushFullPages = true  
)
```

F#

```

new :
    db : uint32 *
    ?page : uint16 *
    ?slot : uint16 *
    ?objectsPerPage : uint16 *
    ?pagesPerDatabase : uint16 *
    ?persistRefs : bool *
    ?tryOtherDatabaseIfLockConflict : bool *
    ?maxNumberOfDatabases : uint32 *
    ?allowOtherTypesOnSamePage : bool *
    ?flushFullPages : bool
(* Defaults:
    let_page = defaultArg page 1
    let_slot = defaultArg slot 1
    let_objectsPerPage = defaultArg objectsPerPage 10000
    let_pagesPerDatabase = defaultArg pagesPerDatabase 10000
    let_persistRefs = defaultArg persistRefs false
    let_tryOtherDatabaseIfLockConflict = defaultArg
tryOtherDatabaseIfLockConflict true
    let_maxNumberOfDatabases = defaultArg maxNumberOfDatabases 4294967295
    let_allowOtherTypesOnSamePage = defaultArg allowOtherTypesOnSamePage
true
    let_flushFullPages = defaultArg flushFullPages true
*)
-> Placement

```

*Parameters**db*Type: [System.UInt32](#)

Database number requested for placement

page (Optional)Type: [System.UInt16](#)

Page number requested for placement

slot (Optional)Type: [System.UInt16](#)

Slot number requested for placement

objectsPerPage (Optional)Type: [System.UInt16](#)

Limit the number of objects per page

pagesPerDatabase (Optional)Type: [System.UInt16](#)

Limit the number of pages per database

persistRefs (Optional)Type: [System.Boolean](#)

VelocityDB Class Library

When persisting an object, shall references be persisted at the same time or later when flushing pages to disk

tryOtherDatabaseIfLockConflict (Optional)

Type: [System.Boolean](#)

Is another Database acceptable

maxNumberOfDatabases (Optional)

Type: [System.UInt32](#)

Maximum number of databases to try for placement

allowOtherTypesOnSamePage (Optional)

Type: [System.Boolean](#)

Is it OK to place object on a page which contains other types

flushFullPages (Optional)

Type: [System.Boolean](#)

Is it OK to flush encountered full pages?

See Also

[Placement Class](#)










[Placement Overload](#)

[VelocityDb Namespace](#)

Placement.Placement Properties

The [Placement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|---------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Gets/sets permission to create multiple types on the same page. |
|  | EndDatabaseNumber | The last Database number acceptable for placement |
|  | MaxObjectsPerPage | Gets/Sets the limit on how many objects to create on a single page. |
|  | MaxPagesPerDatabase | Gets/sets the limit on the number of pages per database. |
|  | StartDatabaseNumber | The first/original Database number used by this Placement |
|  | StartPageNumber | The first/original Page number used by this Placement |
|  | TryDatabaseNumber | Gets the current Database number to try with first for object placement. |
|  | TryPageNumber | Gets/sets the page number to try with first when persisting a new object. |
|  | TrySlotNumber | Gets/sets the page number to try next when persisting an object. |

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.AllowOtherTypesOnSamePage Property

Gets/sets permission to create multiple types on the same page.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AllowOtherTypesOnSamePage { get; set; }
```

VB

```
Public Property AllowOtherTypesOnSamePage As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool AllowOtherTypesOnSamePage {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member AllowOtherTypesOnSamePage : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.EndDatabaseNumber Property

The last Database number acceptable for placement

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint EndDatabaseNumber { get; }
```

VB

```
Public ReadOnly Property EndDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int EndDatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
member EndDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.MaxObjectsPerPage Property

Gets/Sets the limit on how many objects to create on a single page.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort MaxObjectsPerPage { get; set; }
```

VB

```
Public Property MaxObjectsPerPage As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short MaxObjectsPerPage {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member MaxObjectsPerPage : uint16 with get, set
```

Property Value

Type: [UInt16](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.MaxPagesPerDatabase Property

Gets/sets the limit on the number of pages per database.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort MaxPagesPerDatabase { get; set; }
```

VB

```
Public Property MaxPagesPerDatabase As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short MaxPagesPerDatabase {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member MaxPagesPerDatabase : uint16 with get, set
```

Property Value

Type: [UInt16](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.StartDatabaseNumber Property

The first/original Database number used by this Placement

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint StartDatabaseNumber { get; }
```

VB

```
Public ReadOnly Property StartDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int StartDatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
member StartDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.StartPageNumber Property

The first/original Page number used by this Placement

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint StartPageNumber { get; }
```

VB

```
Public ReadOnly Property StartPageNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int StartPageNumber {  
    unsigned int get ();  
}
```

F#

```
member StartPageNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.TryDatabaseNumber Property

Gets the current Database number to try with first for object placement.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint TryDatabaseNumber { get; set; }
```

VB

```
Public Property TryDatabaseNumber As UInteger  
    Get  
    Set
```

C++

```
public:  
property unsigned int TryDatabaseNumber {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
member TryDatabaseNumber : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.TryPageNumber Property

Gets/sets the page number to try with first when persisting a new object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort TryPageNumber { get; set; }
```

VB

```
Public Property TryPageNumber As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short TryPageNumber {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member TryPageNumber : uint16 with get, set
```

Property Value

Type: [UInt16](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.TrySlotNumber Property

Gets/sets the page number to try next when persisting an object.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort TrySlotNumber { get; set; }
```

VB

```
Public Property TrySlotNumber As UShort  
    Get  
    Set
```

C++

```
public:  
property unsigned short TrySlotNumber {  
    unsigned short get ();  
    void set (unsigned short value);  
}
```

F#

```
member TrySlotNumber : uint16 with get, set
```

Property Value

Type: [UInt16](#)

See Also





[Placement Class](#)

[VelocityDb Namespace](#)



Placement.Placement Methods

The [Placement](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IncrementTryDatabaseNumber | Increments the Database number used when placing (persisting) objects |
|  IncrementTryPageNumber | Increments the tryPageNumber by 1 |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase) .) |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.IncrementTryDatabaseNumber Method

Incremnts the Database number used when placing (persisting) objects

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual uint IncrementTryDatabaseNumber ()
```

VB

```
Public Overridable Function IncrementTryDatabaseNumber As UInteger
```

C++

```
public:  
virtual unsigned int IncrementTryDatabaseNumber ()
```

F#

```
abstract IncrementTryDatabaseNumber : unit -> uint32  
override IncrementTryDatabaseNumber : unit -> uint32
```

Return Value

Type: [UInt32](#)

The Database number before incrementing

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.IncrementTryPageNumber Method

Increments the tryPageNumber by 1

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort IncrementTryPageNumber ()
```

VB

```
Public Function IncrementTryPageNumber As UShort
```

C++

```
public:  
unsigned short IncrementTryPageNumber ()
```

F#

```
member IncrementTryPageNumber : unit -> uint16
```

Return Value

Type: [UInt16](#)

The tryPageNumber before the increment

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
    Page^ page,  
    bool useOidShort,  
    Schema^ schema,  
    bool openRefs,  
    List<IOptimizedPersistable^>^ toLoadMembers,  
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
    ) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,V

elocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for

"M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.Placement.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also





[Placement Class](#)

[VelocityDb Namespace](#)

Placement.Placement Fields

The [Placement](#) type exposes the following members.

Fields

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------|
|   | DefaultPlacementDatabaseNumber | If no other Database number is selected when persisting an object then this number is chosen. |
|   | DefaultPlacementPage | If no other Page number is selected when persisting an object then this number is chosen. |

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.DefaultPlacementDatabaseNumber Field

If no other Database number is selected when persisting an object then this number is chosen.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint DefaultPlacementDatabaseNumber = 10
```

VB

```
Public Const DefaultPlacementDatabaseNumber As UInteger = 10
```

C++

```
public:  
literal unsigned int DefaultPlacementDatabaseNumber = 10
```

F#

```
static val mutable DefaultPlacementDatabaseNumber: uint32
```

Field Value

Type: [UInt32](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

Placement.DefaultPlacementPage Field

If no other Page number is selected when persisting an object then this number is chosen.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const ushort DefaultPlacementPage = 1
```

VB

```
Public Const DefaultPlacementPage As UShort = 1
```

C++

```
public:  
literal unsigned short DefaultPlacementPage = 1
```

F#

```
static val mutable DefaultPlacementPage: uint16
```

Field Value

Type: [UInt16](#)

See Also

[Placement Class](#)

[VelocityDb Namespace](#)

ReferenceTracked Class

Tracks references to this object and signals [ReferentialIntegrityException](#) if unpersisted before while still being referenced.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.ReferenceTracked

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReferenceTracked : OptimizedPersistable,
    IReferenceTracked, IOptimizedPersistable, IComparable
```

VB

```
Public Class ReferenceTracked
    Inherits OptimizedPersistable
    Implements IReferenceTracked, IOptimizedPersistable, IComparable
```

C++


```
public ref class ReferenceTracked : public OptimizedPersistable,
    IReferenceTracked, IOptimizedPersistable, IComparable
```

F#


```
type ReferenceTracked =
    class
        inherit OptimizedPersistable
        interface IReferenceTracked
        interface IOptimizedPersistable
        interface IComparable
    end
```

The **ReferenceTracked** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
|  | ReferenceTracked | Initializes a new instance of the ReferenceTracked class |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------|
|  | References | Get a collection of references to this object |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb Namespace](#)

ReferenceTracked Constructor

Initializes a new instance of the [ReferenceTracked](#) class

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReferenceTracked ()
```

VB

```
Public Sub New
```

C++

```
public:  
ReferenceTracked ()
```

F#

```
new : unit -> ReferenceTracked
```

See Also


[ReferenceTracked Class](#)

[VelocityDb Namespace](#)

ReferenceTracked.ReferenceTracked Properties

The [ReferenceTracked](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------|
|  | References | Get a collection of references to this object |

See Also

[ReferenceTracked Class](#)

[VelocityDb Namespace](#)

ReferenceTracked.References Property

Get a collection of references to this object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<Reference> References { get; }
```

VB

```
Public ReadOnly Property References As BTreeSet(Of Reference)  
    Get
```

C++

```
public:  
virtual property BTreeSet<Reference^>^ References {  
    BTreeSet<Reference^>^ get () sealed;  
}
```

F#

```
abstract References : BTreeSet<Reference> with get  
override References : BTreeSet<Reference> with get
```

Property Value

Type: [BTreeSet\(Reference\)](#)

Implements

[IReferenceTracked.References](#)

See Also


[ReferenceTracked Class](#)

[VelocityDb Namespace](#)



ReferenceTracked.ReferenceTracked Methods

The [ReferenceTracked](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[ReferenceTracked Class](#)

[VelocityDb Namespace](#)

ReferenceTracked.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[ReferenceTracked Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T) Class

When a persistent object is opened, all its referenced objects are opened as well. In order to limit the number of opened objects, you need to use this class. A reference from an instance of this class will not cause the referenced object to be opened. It will be opened when you ask for it.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.WeakIOptimizedPersistableReferenceBase](#)

VelocityDb.WeakIOptimizedPersistableReference(T)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class WeakIOptimizedPersistableReference<T> :
WeakIOptimizedPersistableReferenceBase
where T : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class WeakIOptimizedPersistableReference (Of T As
IOptimizedPersistable)
    Inherits WeakIOptimizedPersistableReferenceBase
```

C++

```
[SerializableAttribute]
generic<typename T>
where T : IOptimizedPersistable
public ref class WeakIOptimizedPersistableReference : public
WeakIOptimizedPersistableReferenceBase
```

F#

```
[<SerializableAttribute>]
type WeakIOptimizedPersistableReference<'T when 'T : IOptimizedPersistable> =
    class
        inherit WeakIOptimizedPersistableReferenceBase
    end
```




Type Parameters

T



The [Type](#) of the weak referenced object, must be [OptimizedPersistable](#)

The WeakIOptimizedPersistableReference(T) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  | WeakIOptimizedPersistableReference(T)(UInt64) | Creates a weak reference to some OptimizedPersistable object |
|  | WeakIOptimizedPersistableReference(T)(T) | Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference. |
|  | WeakIOptimizedPersistableReference(T)(Oid) | Creates a weak reference to some OptimizedPersistable object |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------|
|  | Id | Id of weak referenced object (Overrides WeakIOptimizedPersistableReferenceBase.Id.) |
|  | ShortId | Short reference ID of weak reference (Overrides WeakIOptimizedPersistableReferenceBase.ShortId.) |

Methods




| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------|
|  | GetTarget | Gets the persistent object referenced by the weak reference |
|  | ToString | Append the Oid of the referenced object to the string. (Overrides Object.ToString().) |

See Also

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  | WeakIOptimizedPersistableReference(T)(UInt64) | Creates a weak reference to some OptimizedPersistable object |
|  | WeakIOptimizedPersistableReference(T)(T) | Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference. |
|  | WeakIOptimizedPersistableReference(T)(Oid) | Creates a weak reference to some OptimizedPersistable object |

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T) Constructor (UInt64)

Creates a weak reference to some OptimizedPersistable object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReference (  
    ulong oid  
)
```

VB

```
Public Sub New (  
    oid As ULong  
)
```

C++

```
public:  
WeakIOptimizedPersistableReference (  
    unsigned long long oid  
)
```

F#

```
new :  
    oid : uint64 -> WeakIOptimizedPersistableReference
```

Parameters

oid

Type: [System.UInt64](#)

Can be created using `Oid.Encode(databaseNumber, pageNumber, pageNumber)`

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[WeakIOptimizedPersistableReference\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T) Constructor (T)

Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReference (  
    T obj  
)
```

VB

```
Public Sub New (  
    obj As T  
)
```

C++

```
public:  
WeakIOptimizedPersistableReference (  
    T obj  
)
```

F#

```
new :  
    obj : 'T -> WeakIOptimizedPersistableReference
```

Parameters

obj

Type: *T*

[Missing <param name="obj"/> documentation for "M:VelocityDb.WeakIOptimizedPersistableReference`1.#ctor(`0)"]

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[WeakIOptimizedPersistableReference\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T) Constructor (Oid)

Creates a weak reference to some OptimizedPersistable object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReference (  
    Oid oid  
)
```

VB

```
Public Sub New (  
    oid As Oid  
)
```

C++

```
public:  
WeakIOptimizedPersistableReference (  
    Oid oid  
)
```

F#

```
new :  
    oid : Oid -> WeakIOptimizedPersistableReference
```

Parameters

oid

Type: [VelocityDb.Oid](#)

Can be created using `Oid.Encode(databaseNumber, pageNumber, pageNumber)`

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)



[WeakIOptimizedPersistableReference\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T).WeakIOptimizedPersistableReference(T) Properties

The [WeakIOptimizedPersistableReference\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------|
|  | Id | Id of weak referenced object (Overrides WeakIOptimizedPersistableReferenceBase.Id.) |
|  | ShortId | Short reference ID of weak reference (Overrides WeakIOptimizedPersistableReferenceBase.ShortId.) |

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T).Id Property

Id of weak referenced object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Id { get; set; }
```

VB

```
Public Overrides Property Id As ULong  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned long long Id {  
    unsigned long long get () override;  
    void set (unsigned long long value) override;  
}
```

F#

```
abstract Id : uint64 with get, set  
override Id : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[VelocityDb Namespace](#)

WeakOptimizedPersistableReference(T).ShortId Property

Short reference ID of weak reference

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override uint ShortId { get; set; }
```

```
VB  
Public Overrides Property ShortId As UInteger  
    Get  
    Set
```

```
C++  
public:  
virtual property unsigned int ShortId {  
    unsigned int get () override;  
    void set (unsigned int value) override;  
}
```

```
F#  
abstract ShortId : uint32 with get, set  
override ShortId : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also



[WeakOptimizedPersistableReference\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T).WeakIOptimizedPersistableReference(T) Methods

The [WeakIOptimizedPersistableReference\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------|
|  | GetTarget | Gets the persistent object referenced by the weak reference |
|  | ToString | Append the Oid of the referenced object to the string. (Overrides Object.ToString() .) |

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T).GetTarget Method

Gets the persistent object referenced by the weak reference

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T GetTarget(  
    bool update,  
    SessionBase session  
)
```

VB

```
Public Function GetTarget (  
    update As Boolean,  
    session As SessionBase  
) As T
```

C++

```
public:  
T GetTarget(  
    bool update,  
    SessionBase^ session  
)
```

F#

```
member GetTarget :  
    update : bool *  
    session : SessionBase -> 'T
```

Parameters

update

Type: [System.Boolean](#)

Open the persistent object for update?

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing the persistent object

Return Value

Type: *T*

The object referenced by this weak reference

See Also

[WeakIOptimizedPersistableReference\(T\)Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReference(T).ToString Method

Append the Oid of the referenced object to the string.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDb.WeakIOptimizedPersistableReference`1.ToString"]

See Also

[WeakIOptimizedPersistableReference\(T\) Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceBase Class

This is the base class for all weak references.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.WeakIOptimizedPersistableReferenceBase

[VelocityDb.WeakIOptimizedPersistableReference\(T\)](#)

[VelocityDb.WeakIOptimizedPersistableReferenceX\(T\)](#)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class WeakIOptimizedPersistableReferenceBase
```

VB

```
<SerializableAttribute>  
Public MustInherit Class WeakIOptimizedPersistableReferenceBase
```

C++



```
[SerializableAttribute]  
public ref class WeakIOptimizedPersistableReferenceBase abstract
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type WeakIOptimizedPersistableReferenceBase = class end
```

The **WeakIOptimizedPersistableReferenceBase** type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------|
|  | Id | Id of weak referenced object |
|  | ShortId | Short reference ID of weak reference |



See Also

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceBase.WeakIOptimizedPersistableReferenceBase Properties

The [WeakIOptimizedPersistableReferenceBase](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------------------|
|  | Id | Id of weak referenced object |
|  | ShortId | Short reference ID of weak reference |

See Also

[WeakIOptimizedPersistableReferenceBase Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceBase.Id Property

Id of weak referenced object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract ulong Id { get; set; }
```

VB

```
Public MustOverride Property Id As ULong  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned long long Id {  
    unsigned long long get () abstract;  
    void set (unsigned long long value) abstract;  
}
```

F#

```
abstract Id : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also

[WeakIOptimizedPersistableReferenceBase Class](#)

[VelocityDb Namespace](#)

WeakOptimizedPersistableReferenceBase.ShortId Property

Short reference ID of weak reference

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract uint ShortId { get; set; }
```

VB

```
Public MustOverride Property ShortId As UInteger  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned int ShortId {  
    unsigned int get () abstract;  
    void set (unsigned int value) abstract;  
}
```

F#

```
abstract ShortId : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[WeakOptimizedPersistableReferenceBase Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T) Class

Note: This API is now obsolete.

Deprecated, use WeakIOptimizedPersistableReference instead (so name conflict can be avoided) When a persistent object is opened, all its referenced objects are opened as well. In order to limit the number of opened objects, you need to use this class. A reference from an instance of this class will not cause the referenced object to be opened. It will be opened when you ask for it. The [Type](#) of the weak referenced object, must be [OptimizedPersistable](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.WeakIOptimizedPersistableReferenceBase](#)

VelocityDb.WeakIOptimizedPersistableReferenceX(T)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
[ObsoleteAttribute("Use WeakIOptimizedPersistableReference instead (so name
    conflict can be avoided)",
    false)]
public class WeakIOptimizedPersistableReferenceX<T> :
    WeakIOptimizedPersistableReferenceBase
where T : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
<ObsoleteAttribute("Use WeakIOptimizedPersistableReference instead (so name
    conflict can be avoided)",
    false)>
Public Class WeakIOptimizedPersistableReferenceX(Of T As
    IOptimizedPersistable)
    Inherits WeakIOptimizedPersistableReferenceBase
```

C++

```
[SerializableAttribute]
[ObsoleteAttribute(L"Use WeakIOptimizedPersistableReference instead (so name
    conflict can be avoided)",
    false)]
generic<typename T>
where T : IOptimizedPersistable
public ref class WeakIOptimizedPersistableReferenceX : public
    WeakIOptimizedPersistableReferenceBase
```

F#




```
[<SerializableAttribute>]
[<ObsoleteAttribute("Use WeakIOptimizedPersistableReference instead (so name
conflict can be avoided)",
    false)>]
type WeakIOptimizedPersistableReferenceX<'T when 'T : IOptimizedPersistable>
=
    class
        inherit WeakIOptimizedPersistableReferenceBase
    end
```

*Type Parameters**T*



[Missing <typeparam name="T"/> documentation for "T:VelocityDb.WeakIOptimizedPersistableReferenceX`1"]

The WeakIOptimizedPersistableReferenceX(T) type exposes the following members.



Constructors

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  WeakIOptimizedPersistableReferenceX(T)(UInt64) | Creates a weak reference to some OptimizedPersistable object Can be created using <code>Oid.Encode(databaseNumber, pageNumber, pageNumber)</code> |
|  WeakIOptimizedPersistableReferenceX(T)(T) | Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference. |
|  WeakIOptimizedPersistableReferenceX(T)(Oid) | Creates a weak reference to some OptimizedPersistable object Can be created using <code>Oid.Encode(databaseNumber, pageNumber, pageNumber)</code> |

Properties

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|  Id | Id of weak referenced object (Overrides WeakIOptimizedPersistableReferenceBase.Id.) |
|  ShortId | Short reference ID of weak reference (Overrides WeakIOptimizedPersistableReferenceBase.ShortId.) |

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
|  GetTarget | Gets the persistent object referenced by the weak reference |
|  ToString | Append the Oid of the referenced object to the string. (Overrides Object.ToString().) |




VelocityDB Class Library

See Also

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  | WeakIOptimizedPersistableReferenceX(T)(UInt64) | Creates a weak reference to some OptimizedPersistable object Can be created using <code>Oid.Encode(databaseNumber, pageNumber, pageNumber)</code> |
|  | WeakIOptimizedPersistableReferenceX(T)(T) | Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference. |
|  | WeakIOptimizedPersistableReferenceX(T)(Oid) | Creates a weak reference to some OptimizedPersistable object Can be created using <code>Oid.Encode(databaseNumber, pageNumber, pageNumber)</code> |

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T) Constructor (UInt64)

Creates a weak reference to some OptimizedPersistable object Can be created using
Oid.Encode(databaseNumber, pageNumber, pageNumber)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReferenceX(  
    ulong oid  
)
```

VB

```
Public Sub New (  
    oid As ULong  
)
```

C++

```
public:  
WeakIOptimizedPersistableReferenceX(  
    unsigned long long oid  
)
```

F#

```
new :  
    oid : uint64 -> WeakIOptimizedPersistableReferenceX
```

Parameters

oid

Type: [System.UInt64](#)

**[Missing <param name="oid"/> documentation for
"M:VelocityDb.WeakIOptimizedPersistableReferenceX`1.#ctor(System.UInt64)"]**

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[WeakIOptimizedPersistableReferenceX\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T) Constructor (T)

Creates a weak reference to some OptimizedPersistable object. This object must be persisted before adding as a weak reference.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReferenceX(  
    T obj  
)
```

VB

```
Public Sub New (  
    obj As T  
)
```

C++

```
public:  
WeakIOptimizedPersistableReferenceX(  
    T obj  
)
```

F#

```
new :  
    obj : 'T -> WeakIOptimizedPersistableReferenceX
```

Parameters

obj

Type: *T*

[Missing <param name="obj"/> documentation for "M:VelocityDb.WeakIOptimizedPersistableReferenceX`1.#ctor(`0)"]

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[WeakIOptimizedPersistableReferenceX\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T) Constructor (Oid)

Creates a weak reference to some OptimizedPersistable object Can be created using
Oid.Encode(databaseNumber, pageNumber, pageNumber)

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakIOptimizedPersistableReferenceX(  
    Oid oid  
)
```

VB

```
Public Sub New (  
    oid As Oid  
)
```

C++

```
public:  
WeakIOptimizedPersistableReferenceX(  
    Oid oid  
)
```

F#

```
new :  
    oid : Oid -> WeakIOptimizedPersistableReferenceX
```

Parameters

oid

Type: [VelocityDb.Oid](#)

**[Missing <param name="oid"/> documentation for
"M:VelocityDb.WeakIOptimizedPersistableReferenceX`1.#ctor(VelocityDb.Oid)"]**

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)



[WeakIOptimizedPersistableReferenceX\(T\)Overload](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).WeakIOptimizedPersistableReferenceX(T) Properties

The [WeakIOptimizedPersistableReferenceX\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------|
|  | Id | Id of weak referenced object (Overrides WeakIOptimizedPersistableReferenceBase.Id.) |
|  | ShortId | Short reference ID of weak reference (Overrides WeakIOptimizedPersistableReferenceBase.ShortId.) |

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).Id Property

Id of weak referenced object

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Id { get; set; }
```

VB

```
Public Overrides Property Id As ULong  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned long long Id {  
    unsigned long long get () override;  
    void set (unsigned long long value) override;  
}
```

F#

```
abstract Id : uint64 with get, set  
override Id : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).ShortId Property

Short reference ID of weak reference

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint ShortId { get; set; }
```

VB

```
Public Overrides Property ShortId As UInteger  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned int ShortId {  
    unsigned int get () override;  
    void set (unsigned int value) override;  
}
```

F#

```
abstract ShortId : uint32 with get, set  
override ShortId : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also



[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).WeakIOptimizedPersistableReferenceX(T) Methods

The [WeakIOptimizedPersistableReferenceX\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------|
|  | GetTarget | Gets the persistent object referenced by the weak reference |
|  | ToString | Append the Oid of the referenced object to the string. (Overrides Object.ToString() .) |

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).GetTarget Method

Gets the persistent object referenced by the weak reference

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T GetTarget(  
    bool update,  
    SessionBase session  
)
```

VB

```
Public Function GetTarget (  
    update As Boolean,  
    session As SessionBase  
) As T
```

C++

```
public:  
T GetTarget(  
    bool update,  
    SessionBase^ session  
)
```

F#

```
member GetTarget :  
    update : bool *  
    session : SessionBase -> 'T
```

Parameters

update

Type: [System.Boolean](#)

Open the persistent object for update?

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing the persistent object

Return Value

Type: *T*

The object referenced by this weak reference

See Also

[WeakIOptimizedPersistableReferenceX\(T\)Class](#)

VelocityDB Class Library

[VelocityDb Namespace](#)

WeakIOptimizedPersistableReferenceX(T).ToString Method

Append the Oid of the referenced object to the string.

Namespace: [VelocityDb](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDb.WeakIOptimizedPersistableReferenceX`1.ToString"]

See Also














[WeakIOptimizedPersistableReferenceX\(T\) Class](#)

[VelocityDb Namespace](#)


VelocityDb.Collection Namespace

The `VelocityDb.Collection` namespace contains classes for the VelocityDb collections optimized for persistent storage in VelocityDb databases

Classes

| Class | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
|  SortedMap(Key, Value) | A sorted map |
|  SortedMapBase(Key, Value) | A base class for sorted maps. |
|  SortedMapOidShort(Key, Value) | A sorted map where all keys and values and the map are contained within a single Database |
|  SortedMapValueOidShort(Key, Value) | A sorted map where all values are located within a single Database |
|  SortedSetAny(Key) | A sorted set of objects of any Type (not just subclasses of OptimizedPersistable) |
|  SortedSetBase(Key) | A base class for sorted sets |
|  SortedSetOidShort(Key) | A sorted set where the set and all its elements are contained in a single Database |
|  VelocityDbHashSet(T) | A hash code based set |
|  VelocityDbList(T) | A list with an Oid containing objects ordered by index. |
|  VelocityDbListOidShort(T) | List of objects where list and all elements of a list are contained within a single Database |
|  WeakReferenceList(T) | List with an Id containing object Ids ordered by index. |
|  WeakReferenceListBase(T) | List with an Id containing object Ids ordered by index. |
|  WeakShortReferenceList(T) | List with an Id containing object short Ids (page and slot) ordered by index. |

Structures

| Structure | Description |
|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  VelocityDbHashSet(T).Enumerator | Enumerates the elements of a VelocityDbHashSet(Of T) . |

SortedMap(Key, Value) Class

A sorted map

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

[VelocityDb.Collection.SortedMapBase\(Key, Value\)](#)

VelocityDb.Collection.SortedMap(Key, Value)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SortedMap<Key, Value> : SortedMapBase<Key, Value>
```

VB

```
<SerializableAttribute>  
Public Class SortedMap(Of Key, Value)  
    Inherits SortedMapBase(Of Key, Value)
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class SortedMap : public SortedMapBase<Key, Value>
```

F#

```
[<SerializableAttribute>]  
type SortedMap<'Key, 'Value> =  
    class  
        inherit SortedMapBase<'Key, 'Value>  
    end
```

Type Parameters

Key



Element key type

Value




Element value type

The SortedMap(Key, Value) type exposes the following members.


Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------|
|  | SortedMap(Key, Value)(Int32) | Constructs a new map |
|  | SortedMap(Key, Value)(VelocityDbComparer(Key)) | Constructs a new map with a given comparator |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the Key comparator of the map (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal list of NodeKeys (Overrides SortedSetBase(Key).Keys.) |
|  | Values | Gets the internal list of NodeValues (Overrides SortedMapBase(Key, Value).Values.) |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)).) |

Extension Methods



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------|
|  | SortedMap(Key, Value)(Int32) | Constructs a new map |
|  | SortedMap(Key, Value)(VelocityDbComparer(Key)) | Constructs a new map with a given comparator |

See Also

[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value) Constructor (Int32)

Constructs a new map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMap(  
    int capacity = 0  
)
```

VB

```
Public Sub New (  
    Optional capacity As Integer = 0  
)
```

C++

```
public:  
SortedMap(  
    int capacity = 0  
)
```

F#

```
new :  
    ?capacity : int  
(* Defaults:  
    let_capacity = defaultArg capacity 0  
)  
-> SortedMap
```

Parameters

capacity (Optional)

Type: [System.Int32](#)

Resizes internal list to a requested size

See Also

[SortedMap\(Key, Value\)Class](#)

[SortedMap\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value) Constructor (VelocityDbComparer(Key))

Constructs a new map with a given comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMap(  
    VelocityDbComparer<Key> comparer  
)
```

VB

```
Public Sub New (  
    comparer As VelocityDbComparer(Of Key)  
)
```

C++

```
public:  
SortedMap(  
    VelocityDbComparer<Key>^ comparer  
)
```

F#

```
new :  
    comparer : VelocityDbComparer<'Key> -> SortedMap
```

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

The comparator is referenced by a OidShort so it must exist within the same Database as the SortedMap

See Also

[SortedMap\(Key, Value\)Class](#)




[SortedMap\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value).SortedMap(Key, Value) Properties

The [SortedMap\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the Key comparator of the map (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal list of NodeKeys (Overrides SortedSetBase(Key).Keys.) |
|  | Values | Gets the internal list of NodeValues (Overrides SortedMapBase(Key, Value).Values.) |

See Also

[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value).Comparer Property

Gets the Key comparator of the map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value).Keys Property

Gets the internal list of NodeKeys

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Key> Keys { get; }
```

VB

```
Public Overrides ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () override;  
}
```

F#

```
abstract Keys : IList<'Key> with get  
override Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also

[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value).Values Property

Gets the internal list of NodeValues

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Value> Values { get; }
```

VB

```
Public Overrides ReadOnly Property Values As IList(Of Value)  
    Get
```

C++

```
public:  
virtual property IList<Value>^ Values {  
    IList<Value>^ get () override;  
}
```

F#

```
abstract Values : IList<'Value> with get  
override Values : IList<'Value> with get
```

Property Value

Type: [IList\(Value\)](#)

See Also


[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)



SortedMap(Key, Value).SortedMap(Key, Value) Methods

The [SortedMap\(Key, Value\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[SortedMap\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMap(Key, Value).Persist Method

Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
```

```
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of this persistent object

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue<IOptimizedPersistable>\)](#)

See Also

[SortedMap<Key, Value>Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value) Class

A base class for sorted maps.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

VelocityDb.Collection.SortedMapBase(Key, Value)

[VelocityDb.Collection.SortedMap\(Key, Value\)](#)

[VelocityDb.Collection.SortedMapOidShort\(Key, Value\)](#)

[VelocityDb.Collection.SortedMapValueOidShort\(Key, Value\)](#)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class SortedMapBase<Key, Value> : SortedSetBase<Key>,  
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable
```

VB

```
<SerializableAttribute>  
Public MustInherit Class SortedMapBase(Of Key, Value)  
    Inherits SortedSetBase(Of Key)  
    Implements IEnumerable(Of KeyValuePair(Of Key, Value)),  
    IEnumerable
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class SortedMapBase abstract : public SortedSetBase<Key>,  
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type SortedMapBase<'Key, 'Value> =  
    class  
        inherit SortedSetBase<'Key>  
        interface IEnumerable<KeyValuePair<'Key, 'Value>>  
        interface IEnumerable  
    end
```

*Type Parameters***Key**



A key type for a map

Value











A value type for a map

The SortedMapBase(Key, Value) type exposes the following members.



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|----------------------------------------------|
|  | Item | Gets or sets the value associated with a key |
|  | Values | The map values as a list |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | Add(KeyValuePair(Key, Value)) | Add to a map |
|  | Add(Key, Value) | Add to a map |
|  | Clear | Removes all keys and values from the map, leaving the map empty (Overrides SortedSetBase(Key).Clear() .) |
|  | Contains | Check if map contains a key value pair |
|  | ContainsKey | Checks if map contains a key |
|  | CopyTo | throw new NotImplementedException(); |
|  | GetEnumerator | |
|  | IsEmpty | Check if map is empty |
|  | Remove(KeyValuePair(Key, Value)) | Removes a key value pair from the map (if found) |
|  | Remove(Key) | Removes a key from the map (if found) (Overrides SortedSetBase(Key).Remove(Key) .) |
|  | TryGetValue | Gets the value associated with the specified key. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |



See Also

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).SortedMapBase(Key, Value) Properties

The [SortedMapBase\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|----------------------------------------------|
|  | Item | Gets or sets the value associated with a key |
|  | Values | The map values as a list |

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Item Property

Gets or sets the value associated with a key

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Value this[
    Key key
] { get; set; }
```

VB

```
Public Default Property Item (
    key As Key
) As Value
    Get
    Set
```

C++

```
public:
property Value default[Key key] {
    Value get (Key key);
    void set (Key key, Value value);
}
```

F#

```
member Item : 'Value with get, set
```

Parameters

key

Type: *Key*

The lookup key

Return Value

Type: *Value*

The value associated with a key or throws *KeyNotFoundException*

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Values Property

The map values as a list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IList<Value> Values { get; }
```

VB

```
Public MustOverride ReadOnly Property Values As IList(Of Value)  
    Get
```

C++

```
public:  
virtual property IList<Value>^ Values {  
    IList<Value>^ get () abstract;  
}
```

F#

```
abstract Values : IList<'Value> with get
```

Property Value

Type: [IList\(Value\)](#)

See Also












[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)



SortedMapBase(Key, Value).SortedMapBase(Key, Value) Methods

The [SortedMapBase\(Key, Value\)](#) generic type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  Add(KeyValuePair(Key, Value)) | Add to a map |
|  Add(Key, Value) | Add to a map |
|  Clear | Removes all keys and values from the map, leaving the map empty (Overrides SortedSetBase(Key).Clear() .) |
|  Contains | Check if map contains a key value pair |
|  ContainsKey | Checks if map contains a key |
|  CopyTo | throw new NotImplementedException(); |
|  GetEnumerator | |
|  IsEmpty | Check if map is empty |
|  Remove(KeyValuePair(Key, Value)) | Removes a key value pair from the map (if found) |
|  Remove(Key) | Removes a key from the map (if found) (Overrides SortedSetBase(Key).Remove(Key) .) |
|  TryGetValue | Gets the value associated with the specified key. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Add Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|--------------|
|  | Add(KeyValuePair(Key, Value)) | Add to a map |
|  | Add(Key, Value) | Add to a map |

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Add Method (KeyValuePair(Key, Value))

Add to a map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Add(  
    KeyValuePair<Key, Value> keyValue  
)
```

VB

```
Public Sub Add (  
    keyValue As KeyValuePair(Of Key, Value)  
)
```

C++

```
public:  
void Add(  
    KeyValuePair<Key, Value> keyValue  
)
```

F#

```
member Add :  
    keyValue : KeyValuePair<'Key, 'Value> -> unit
```

Parameters

keyValue

Type: [System.Collections.Generic.KeyValuePair](#)(Key, Value)

key and value being added

See Also

[SortedMapBase\(Key, Value\)Class](#)

[Add Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Add Method (Key, Value)

Add to a map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Add(  
    Key key,  
    Value value  
)
```

VB

```
Public Sub Add (  
    key As Key,  
    value As Value  
)
```

C++

```
public:  
void Add(  
    Key key,  
    Value value  
)
```

F#

```
member Add :  
    key : 'Key *  
    value : 'Value -> unit
```

Parameters

key

Type: *Key*

key added

value

Type: *Value*

value added

See Also

[SortedMapBase\(Key, Value\)Class](#)

[Add Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Clear Method

Removes all keys and values from the map, leaving the map empty

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Contains Method

Check if map contains a key value pair

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    KeyValuePair<Key, Value> keyValue  
)
```

VB

```
Public Function Contains (  
    keyValue As KeyValuePair(Of Key, Value)  
) As Boolean
```

C++

```
public:  
bool Contains(  
    KeyValuePair<Key, Value> keyValue  
)
```

F#

```
member Contains :  
    keyValue : KeyValuePair<'Key, 'Value> -> bool
```

Parameters

keyValue

Type: [System.Collections.Generic.KeyValuePair](#)(Key, Value)

[Missing <param name="keyValue"/> documentation for

"M:VelocityDb.Collection.SortedMapBase`2.Contains(System.Collections.Generic.KeyValuePair{`0,`1})"

Return Value

Type: [Boolean](#)

true if map contains the key value pair

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).ContainsKey Method

Checks if map contains a key

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ContainsKey(  
    Key aKey  
)
```

VB

```
Public Function ContainsKey (  
    aKey As Key  
) As Boolean
```

C++

```
public:  
bool ContainsKey(  
    Key aKey  
)
```

F#

```
member ContainsKey :  
    aKey : 'Key -> bool
```

Parameters

aKey

Type: *Key*

[Missing <param name="aKey"/> documentation for "M:VelocityDb.Collection.SortedMapBase`2.ContainsKey(`0)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.SortedMapBase`2.ContainsKey(`0)"]

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).CopyTo Method

throw new NotImplementedException();

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo (
    KeyValuePair<Key, Value>[] array,
    int index
)
```

VB

```
Public Sub CopyTo (
    array As KeyValuePair(Of Key, Value) (),
    index As Integer
)
```

C++

```
public:
void CopyTo (
    array<KeyValuePair<Key, Value>>^ array,
    int index
)
```

F#

```
member CopyTo :
    array : KeyValuePair<'Key, 'Value>[] *
    index : int -> unit
```

Parameters

array

Type: [System.Collections.Generic.KeyValuePair\(Key, Value\)](#)[]

The destination array

index

Type: [System.Int32](#)

The start index

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).GetEnumerator Method

[Missing <summary> documentation for "M:VelocityDb.Collection.SortedMapBase`2.GetEnumerator"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<KeyValuePair<Key, Value>> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of KeyValuePair(Of Key, Value))
```

C++

```
public:  
virtual IEnumerable<KeyValuePair<Key, Value>>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<KeyValuePair<'Key, 'Value>>  
override GetEnumerator : unit -> IEnumerable<KeyValuePair<'Key, 'Value>>
```

Return Value

Type: [IEnumerable\(KeyValuePair\(Key, Value\)\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.SortedMapBase`2.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[SortedMapBase\(Key, Value\) Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).isEmpty Method

Check if map is empty

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool isEmpty()
```

VB

```
Public Function isEmpty As Boolean
```

C++

```
public:  
bool isEmpty()
```

F#

```
member isEmpty : unit -> bool
```

Return Value

Type: [Boolean](#)

true if map is empty; otherwise false



See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Remove Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------------|
|  | Remove(KeyValuePair(Key, Value)) | Removes a key value pair from the map (if found) |
|  | Remove(Key) | Removes a key from the map (if found) (Overrides SortedSetBase(Key).Remove(Key).) |

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Remove Method (KeyValuePair(Key, Value))

Removes a key value pair from the map (if found)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Remove(  
    KeyValuePair<Key, Value> keyValue  
)
```

VB

```
Public Function Remove (  
    keyValue As KeyValuePair(Of Key, Value)  
) As Boolean
```

C++

```
public:  
bool Remove(  
    KeyValuePair<Key, Value> keyValue  
)
```

F#

```
member Remove :  
    keyValue : KeyValuePair<'Key, 'Value> -> bool
```

Parameters

keyValue

Type: [System.Collections.Generic.KeyValuePair\(Key, Value\)](#)

[Missing <param name="keyValue"/> documentation for "M:VelocityDb.Collection.SortedMapBase`2.Remove(System.Collections.Generic.KeyValuePair{`0,`1})"]

Return Value

Type: [Boolean](#)

true if key was found and removed; otherwise false

See Also

[SortedMapBase\(Key, Value\)Class](#)

[Remove Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).Remove Method (Key)

Removes a key from the map (if found)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Remove (  
    Key key  
)
```

VB

```
Public Overrides Function Remove (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    Key key  
) override
```

F#

```
abstract Remove :  
    key : 'Key -> bool  
override Remove :  
    key : 'Key -> bool
```

Parameters

key

Type: Key

[Missing <param name="key"/> documentation for "M:VelocityDb.Collection.SortedMapBase`2.Remove(`0)"]

Return Value

Type: [Boolean](#)

true if key was found and removed; otherwise false

See Also

[SortedMapBase\(Key, Value\)Class](#)

[Remove Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapBase(Key, Value).TryGetValue Method

Gets the value associated with the specified key.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetValue(  
    Key key,  
    out Value value  
)
```

VB

```
Public Function TryGetValue (  
    key As Key,  
    <OutAttribute> ByRef value As Value  
) As Boolean
```

C++

```
public:  
bool TryGetValue(  
    Key key,  
    [OutAttribute] Value% value  
)
```

F#

```
member TryGetValue :  
    key : 'Key *  
    value : 'Value byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

value

Type: *Value*

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the set contains an element with the specified key; otherwise, false.

VelocityDB Class Library

See Also

[SortedMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value) Class

A sorted map where all keys and values and the map are contained within a single [Database](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

[VelocityDb.Collection.SortedMapBase\(Key, Value\)](#)

VelocityDb.Collection.SortedMapOidShort(Key, Value)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SortedMapOidShort<Key, Value> : SortedMapBase<Key, Value>  
where Key : OptimizedPersistable  
where Value : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class SortedMapOidShort(Of Key As OptimizedPersistable, Value As  
OptimizedPersistable)  
    Inherits SortedMapBase(Of Key, Value)
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
where Key : OptimizedPersistable  
where Value : OptimizedPersistable  
public ref class SortedMapOidShort : public SortedMapBase<Key, Value>
```

F#

```
[<SerializableAttribute>]  
type SortedMapOidShort<'Key, 'Value when 'Key : OptimizedPersistable when  
'Value : OptimizedPersistable> =  
    class  
        inherit SortedMapBase<'Key, 'Value>  
    end
```

Type Parameters

Key



The Type of keys of this map

Value





The Type of values of this map

The SortedMapOidShort(Key, Value) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------|
|  | SortedMapOidShort(Key, Value)() | Constructs a new empty map |
|  | SortedMapOidShort(Key, Value)(VelocityDbComparer(Key)) | Constructs a new empty map with a given object comparator |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------|
|  | Comparer | Gets comparator (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal list of keys (Overrides SortedSetBase(Key).Keys.) |
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  | Values | Gets the internal list of values (Overrides SortedMapBase(Key, Value).Values.) |

Extension Methods



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------|-----------------------------------------------------------|
|  | SortedMapOidShort(Key, Value)() | Constructs a new empty map |
|  | SortedMapOidShort(Key, Value)(VelocityDbComparer(Key)) | Constructs a new empty map with a given object comparator |

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value) Constructor

Constructs a new empty map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMapOidShort ()
```

VB

```
Public Sub New
```

C++

```
public:  
SortedMapOidShort ()
```

F#

```
new : unit -> SortedMapOidShort
```

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[SortedMapOidShort\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value) Constructor (VelocityDbComparer(Key))

Constructs a new empty map with a given object comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMapOidShort (  
    VelocityDbComparer<Key> comparer  
)
```

VB

```
Public Sub New (  
    comparer As VelocityDbComparer(Of Key)  
)
```

C++

```
public:  
SortedMapOidShort (  
    VelocityDbComparer<Key>^ comparer  
)
```

F#

```
new :  
    comparer : VelocityDbComparer<'Key> -> SortedMapOidShort
```

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[SortedMapOidShort\(Key, Value\)Overload](#)





[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value).SortedMapOidShort(Key, Value)

Properties

The [SortedMapOidShort\(Key, Value\)](#) generic type exposes the following members.

Properties

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  Comparer | Gets comparator (Overrides SortedSetBase(Key).Comparer.) |
|  Keys | Gets the internal list of keys (Overrides SortedSetBase(Key).Keys.) |
|  MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  Values | Gets the internal list of values (Overrides SortedMapBase(Key, Value).Values.) |

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value).Comparer Property

Gets comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value).Keys Property

Gets the internal list of keys

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Key> Keys { get; }
```

VB

```
Public Overrides ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () override;  
}
```

F#

```
abstract Keys : IList<'Key> with get  
override Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value).MaxNumberOfDatabases Property

[Missing <summary> documentation for

"P:VelocityDb.Collection.SortedMapOidShort`2.MaxNumberOfDatabases"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also

[SortedMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapOidShort(Key, Value).Values Property

Gets the internal list of values

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Value> Values { get; }
```

VB

```
Public Overrides ReadOnly Property Values As IList(Of Value)  
    Get
```

C++

```
public:  
virtual property IList<Value>^ Values {  
    IList<Value>^ get () override;  
}
```

F#

```
abstract Values : IList<'Value> with get  
override Values : IList<'Value> with get
```

Property Value

Type: [IList\(Value\)](#)

See Also

[SortedMapOidShort\(Key, Value\)Class](#)



[VelocityDb.Collection Namespace](#)

SortedMapOidShort(*Key*, *Value*).SortedMapOidShort(*Key*, *Value*)

Methods

The [SortedMapOidShort\(*Key*, *Value*\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[SortedMapOidShort\(*Key*, *Value*\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value) Class

A sorted map where all values are located within a single [Database](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

[VelocityDb.Collection.SortedMapBase\(Key, Value\)](#)

VelocityDb.Collection.SortedMapValueOidShort(Key, Value)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SortedMapValueOidShort<Key, Value> : SortedMapBase<Key, Value>  
where Value : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class SortedMapValueOidShort(Of Key, Value As OptimizedPersistable)  
    Inherits SortedMapBase(Of Key, Value)
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
where Value : OptimizedPersistable  
public ref class SortedMapValueOidShort : public SortedMapBase<Key, Value>
```

F#

```
[<SerializableAttribute>]  
type SortedMapValueOidShort<'Key, 'Value when 'Value : OptimizedPersistable>  
=  
    class  
        inherit SortedMapBase<'Key, 'Value>  
    end
```




Type Parameters

Key





Value

The SortedMapValueOidShort(Key, Value) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------|
|  | SortedMapValueOidShort(Key, Value)() | Constructs a new map with OidShort based values |
|  | SortedMapValueOidShort(Key, Value)(Int32) | Constructs a new map with OidShort based values |
|  | SortedMapValueOidShort(Key, Value)(VelocityDbComparer(Key)) | Constructs a new map with a given comparator |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the Key comparator of the map (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal list of NodeKeys (Overrides SortedSetBase(Key).Keys.) |
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  | Values | Gets the internal list of NodeValues (Overrides SortedMapBase(Key, Value).Values.) |

Extension Methods




| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------|
|  | SortedMapValueOidShort(Key, Value)() | Constructs a new map with OidShort based values |
|  | SortedMapValueOidShort(Key, Value)(Int32) | Constructs a new map with OidShort based values |
|  | SortedMapValueOidShort(Key, Value)(VelocityDbComparer(Key)) | Constructs a new map with a given comparator |

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value) Constructor

Constructs a new map with OidShort based values

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMapValueOidShort ()
```

VB

```
Public Sub New
```

C++

```
public:  
SortedMapValueOidShort ()
```

F#

```
new : unit -> SortedMapValueOidShort
```

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[SortedMapValueOidShort\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value) Constructor (Int32)

Constructs a new map with OidShort based values

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMapValueOidShort (  
    int capacity  
)
```

VB

```
Public Sub New (  
    capacity As Integer  
)
```

C++

```
public:  
SortedMapValueOidShort (  
    int capacity  
)
```

F#

```
new :  
    capacity : int -> SortedMapValueOidShort
```

Parameters

capacity

Type: [System.Int32](#)

Presizes internal list to a requested size

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[SortedMapValueOidShort\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value) Constructor (VelocityDbComparer(Key))

Constructs a new map with a given comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedMapValueOidShort (  
    VelocityDbComparer<Key> comparer  
)
```

VB

```
Public Sub New (  
    comparer As VelocityDbComparer(Of Key)  
)
```

C++

```
public:  
SortedMapValueOidShort (  
    VelocityDbComparer<Key>^ comparer  
)
```

F#

```
new :  
    comparer : VelocityDbComparer<'Key> -> SortedMapValueOidShort
```

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)





[SortedMapValueOidShort\(Key, Value\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).SortedMapValueOidShort(Key, Value) Properties

The [SortedMapValueOidShort\(Key, Value\)](#) generic type exposes the following members.

Properties

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|  Comparer | Gets the Key comparator of the map (Overrides SortedSetBase(Key).Comparer.) |
|  Keys | Gets the internal list of NodeKeys (Overrides SortedSetBase(Key).Keys.) |
|  MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  Values | Gets the internal list of NodeValues (Overrides SortedMapBase(Key, Value).Values.) |

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).Comparer Property

Gets the Key comparator of the map

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).Keys Property

Gets the internal list of NodeKeys

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Key> Keys { get; }
```

VB

```
Public Overrides ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () override;  
}
```

F#

```
abstract Keys : IList<'Key> with get  
override Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).MaxNumberOfDatabases Property

[Missing <summary> documentation for

"P:VelocityDb.Collection.SortedMapValueOidShort`2.MaxNumberOfDatabases"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).Values Property

Gets the internal list of NodeValues

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Value> Values { get; }
```

VB

```
Public Overrides ReadOnly Property Values As IList(Of Value)  
    Get
```

C++

```
public:  
virtual property IList<Value>^ Values {  
    IList<Value>^ get () override;  
}
```

F#

```
abstract Values : IList<'Value> with get  
override Values : IList<'Value> with get
```

Property Value

Type: [IList\(Value\)](#)

See Also



[SortedMapValueOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedMapValueOidShort(Key, Value).SortedMapValueOidShort(Key, Value) Methods

The [SortedMapValueOidShort\(Key, Value\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[SortedMapValueOidShort\(Key, Value\)Class](#)
[VelocityDb.Collection Namespace](#)

SortedSetAny(Key) Class

A sorted set of objects of any Type (not just subclasses of [OptimizedPersistable](#))

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

VelocityDb.Collection.SortedSetAny(Key)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class SortedSetAny<Key> : SortedSetBase<Key>,
    ISet<Key>, ICollection<Key>, IEnumerable<Key>, IEnumerable
```

VB

```
<SerializableAttribute>
Public Class SortedSetAny(Of Key)
    Inherits SortedSetBase(Of Key)
    Implements ISet(Of Key), ICollection(Of Key),
        IEnumerable(Of Key), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename Key>
public ref class SortedSetAny : public SortedSetBase<Key>,
    ISet<Key>, ICollection<Key>, IEnumerable<Key>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type SortedSetAny<'Key> =
    class
        inherit SortedSetBase<'Key>
        interface ISet<'Key>
        interface ICollection<'Key>
        interface IEnumerable<'Key>
        interface IEnumerable
    end
```




Type Parameters

Key



The type of element for this list.

The SortedSetAny(Key) type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------|
|  | SortedSetAny(Key)() | Constructs the set |
|  | SortedSetAny(Key)(Int32) | Constructs a set with a given initial capacity |
|  | SortedSetAny(Key)(VelocityDbComparer(Key)) | Constructs a set with a given object comparator |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------|
|  | Comparer | Gets the set comparator (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal Key list of the set (Overrides SortedSetBase(Key).Keys.) |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)).) |

Extension Methods




| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------|
|  | SortedSetAny(Key)() | Constructs the set |
|  | SortedSetAny(Key)(Int32) | Constructs a set with a given initial capacity |
|  | SortedSetAny(Key)(VelocityDbComparer(Key)) | Constructs a set with a given object comparator |

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key) Constructor

Constructs the set

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetAny ()
```

VB

```
Public Sub New
```

C++

```
public:  
SortedSetAny ()
```

F#

```
new : unit -> SortedSetAny
```

See Also

[SortedSetAny\(Key\)Class](#)

[SortedSetAny\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key) Constructor (Int32)

Constructs a set with a given initial capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetAny(  
    int capacity  
)
```

VB

```
Public Sub New (  
    capacity As Integer  
)
```

C++

```
public:  
SortedSetAny(  
    int capacity  
)
```

F#

```
new :  
    capacity : int -> SortedSetAny
```

Parameters

capacity

Type: [System.Int32](#)

See Also

[SortedSetAny\(Key\)Class](#)

[SortedSetAny\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key) Constructor (VelocityDbComparer(Key))

Constructs a set with a given object comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetAny(  
    VelocityDbComparer<Key> comparer  
)
```

VB

```
Public Sub New (  
    comparer As VelocityDbComparer(Of Key)  
)
```

C++

```
public:  
SortedSetAny(  
    VelocityDbComparer<Key>^ comparer  
)
```

F#

```
new :  
    comparer : VelocityDbComparer<'Key> -> SortedSetAny
```

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

A comparator of Key objects

See Also

[SortedSetAny\(Key\)Class](#)



[SortedSetAny\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key).SortedSetAny(Key) Properties

The [SortedSetAny\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------------------------------------------|
|  | Comparer | Gets the set comparator (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal Key list of the set (Overrides SortedSetBase(Key).Keys.) |

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key).Comparer Property

Gets the set comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key).Keys Property

Gets the internal Key list of the set

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Key> Keys { get; }
```

VB

```
Public Overrides ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () override;  
}
```

F#

```
abstract Keys : IList<'Key> with get  
override Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)



SortedSetAny(Key).SortedSetAny(Key) Methods

The [SortedSetAny\(Key\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetAny(Key).Persist Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _persistRefs = defaultArg persistRefs true
    let _disableFlush = defaultArg disableFlush false
    let _toPersist = defaultArg toPersist null
```

```
*)
-> uint64
override Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

[Missing <param name="place"/> documentation for "M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

persistRefs (Optional)

Type: [System.Boolean](#)

[Missing <param name="persistRefs"/> documentation for "M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

disableFlush (Optional)

Type: [System.Boolean](#)

[Missing <param name="disableFlush"/> documentation for "M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

[Missing <param name="toPersist"/> documentation for "M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBa

se,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.SortedSetAny`1.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue\(IOptimizedPersistable\)\)](#)

See Also

[SortedSetAny\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key) Class

A base class for sorted sets

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

[VelocityDb.Collection.SortedMapBase\(Key, Value\)](#)

[VelocityDb.Collection.SortedSetAny\(Key\)](#)

[VelocityDb.Collection.SortedSetOidShort\(Key\)](#)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class SortedSetBase<Key> : OptimizedPersistable,  
    IEnumerable<Key>, IEnumerable
```

VB

```
<SerializableAttribute>  
Public MustInherit Class SortedSetBase(Of Key)  
    Inherits OptimizedPersistable  
    Implements IEnumerable(Of Key), IEnumerable
```

C++

```
[SerializableAttribute]  
generic<typename Key>  
public ref class SortedSetBase abstract : public OptimizedPersistable,  
    IEnumerable<Key>, IEnumerable
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type SortedSetBase<'Key> =  
    class  
        inherit OptimizedPersistable  
        interface IEnumerable<'Key>  
        interface IEnumerable  
    end
```







Type Parameters

Key









A key type for a set.

The SortedSetBase(Key) type exposes the following members.



Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Comparer | Gets the IComparer that is used to determine the order of the items in the sorted set |
|  Count | Number of elements in this SortedSetBase |
|  IsEmpty | Returns true if 0 elements are in the set. |
|  IsReadOnly | Returns false |
|  Item | Given a lookup key, returns the corresponding key within the collection or null if not found |
|  Keys | Gets the internal list of NodeKeys |

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
|  Add | Add item to this sorted set. |
|  Clear | Remove all items from this set. |
|  Contains | Checks if this sorted set contains the item |
|  CopyTo | Copy items in this sorted set to array, starting at arrayIndex |
|  IndexOf | Get the index of a given key |
|  Remove | Remove item from this set |
|  RemoveWhere | Remove elements that match specified predicate. Returns the number of elements removed |
|  TryGetValue | Gets the value associated with the specified key. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |







See Also

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).SortedSetBase(Key) Properties

The [SortedSetBase\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|----------------------------------------------------------------------------------------------|
|  | Comparer | Gets the IComparer that is used to determine the order of the items in the sorted set |
|  | Count | Number of elements in this SortedSetBase |
|  | IsEmpty | Returns true if 0 elements are in the set. |
|  | IsReadOnly | Returns false |
|  | Item | Given a lookup key, returns the corresponding key within the collection or null if not found |
|  | Keys | Gets the internal list of NodeKeys |

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Comparer Property

Gets the IComparer that is used to determine the order of the items in the sorted set

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public MustOverride ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () abstract;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Count Property

Number of elements in this SortedSetBase

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).IsEmpty Property

Returns true if 0 elements are in the set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsEmpty { get; }
```

VB

```
Public ReadOnly Property IsEmpty As Boolean  
    Get
```

C++

```
public:  
property bool IsEmpty {  
    bool get ();  
}
```

F#

```
member IsEmpty : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).IsReadOnly Property

Returns false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsReadOnly { get; }
```

VB

```
Public ReadOnly Property IsReadOnly As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsReadOnly {  
    bool get () sealed;  
}
```

F#

```
abstract IsReadOnly : bool with get  
override IsReadOnly : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Item Property

Given a lookup key, returns the corresponding key within the collection or null if not found

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key this[
    Key key
] { get; set; }
```

VB

```
Public Default Property Item (
    key As Key
) As Key
    Get
    Set
```

C++

```
public:
property Key default[Key key] {
    Key get (Key key);
    void set (Key key, Key value);
}
```

F#

```
member Item : 'Key with get, set
```

Parameters

key

Type: *Key*

The lookup key

Return Value

Type: *Key*

The key matching the lookup key or null if not found

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Keys Property

Gets the internal list of NodeKeys

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IList<Key> Keys { get; }
```

VB

```
Public MustOverride ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () abstract;  
}
```

F#

```
abstract Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also









[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)



SortedSetBase(Key).SortedSetBase(Key) Methods

The [SortedSetBase\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
|  Add | Add item to this sorted set. |
|  Clear | Remove all items from this set. |
|  Contains | Checks if this sorted set contains the item |
|  CopyTo | Copy items in this sorted set to array, starting at arrayIndex |
|  IndexOf | Get the index of a given key |
|  Remove | Remove item from this set |
|  RemoveWhere | Remove elements that match specified predicate. Returns the number of elements removed |
|  TryGetValue | Gets the value associated with the specified key. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Add Method

Add item to this sorted set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Add(  
    Key key  
)
```

VB

```
Public Function Add (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Add(  
    Key key  
) sealed
```

F#

```
abstract Add :  
    key : 'Key -> bool  
override Add :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

item to add

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.SortedSetBase`1.Add(`0)"]

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Clear Method

Remove all items from this set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Clear ()
```

VB

```
Public Overridable Sub Clear
```

C++

```
public:  
virtual void Clear ()
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Contains Method

Checks if this sorted set contains the item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Key aKey  
)
```

VB

```
Public Function Contains (  
    aKey As Key  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Key aKey  
) sealed
```

F#

```
abstract Contains :  
    aKey : 'Key -> bool  
override Contains :  
    aKey : 'Key -> bool
```

Parameters

aKey

Type: *Key*

item to check for containment

Return Value

Type: [Boolean](#)

true if item contained; false if not

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).CopyTo Method

Copy items in this sorted set to array, starting at arrayIndex

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    Key[] array,  
    int arrayIndex,  
    int count  
)
```

VB

```
Public Sub CopyTo (  
    array As Key(),  
    arrayIndex As Integer,  
    count As Integer  
)
```

C++

```
public:  
void CopyTo(  
    array<Key>^ array,  
    int arrayIndex,  
    int count  
)
```

F#

```
member CopyTo :  
    array : 'Key[] *  
    arrayIndex : int *  
    count : int -> unit
```

Parameters

array

Type: [Key\[\]](#)

array to add items to

arrayIndex

Type: [System.Int32](#)

index to start at

count

Type: [System.Int32](#)

number of elements to copy

VelocityDB Class Library

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).IndexOf Method

Get the index of a given key

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int IndexOf(  
    Key aKey  
)
```

VB

```
Public Function IndexOf (  
    aKey As Key  
) As Integer
```

C++

```
public:  
int IndexOf(  
    Key aKey  
)
```

F#

```
member IndexOf :  
    aKey : 'Key -> int
```

Parameters

aKey

Type: *Key*

The lookup key

Return Value

Type: [Int32](#)

The index of key or -1 if a matching key was not found.

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).Remove Method

Remove item from this set

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool Remove (  
    Key key  
)
```

VB

```
Public Overridable Function Remove (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    Key key  
)
```

F#

```
abstract Remove :  
    key : 'Key -> bool  
override Remove :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

item to remove

Return Value

Type: [Boolean](#)

true if removed; false if not (i.e. if the item wasn't in the SortedSetBase)

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).RemoveWhere Method

Remove elements that match specified predicate. Returns the number of elements removed

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int RemoveWhere(  
    Predicate<Key> match  
)
```

VB

```
Public Function RemoveWhere (  
    match As Predicate(Of Key)  
) As Integer
```

C++

```
public:  
int RemoveWhere(  
    Predicate<Key>^ match  
)
```

F#

```
member RemoveWhere :  
    match : Predicate<'Key> -> int
```

Parameters

match

Type: [System.Predicate\(Key\)](#)

[Missing <param name="match"/> documentation for

"M:VelocityDb.Collection.SortedSetBase`1.RemoveWhere(System.Predicate{`0})"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.SortedSetBase`1.RemoveWhere(System.Predicate{`0})"]

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetBase(Key).TryGetValue Method

Gets the value associated with the specified key.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetValue(  
    Key key,  
    ref Key value  
)
```

VB

```
Public Function TryGetValue (  
    key As Key,  
    ByRef value As Key  
) As Boolean
```

C++

```
public:  
bool TryGetValue(  
    Key key,  
    Key% value  
)
```

F#

```
member TryGetValue :  
    key : 'Key *  
    value : 'Key byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

value

Type: *Key*

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the set contains an element with the specified key; otherwise, false.

VelocityDB Class Library

See Also

[SortedSetBase\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key) Class

A sorted set where the set and all its elements are contained in a single [Database](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.SortedSetBase\(Key\)](#)

VelocityDb.Collection.SortedSetOidShort(Key)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SortedSetOidShort<Key> : SortedSetBase<Key>  
where Key : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class SortedSetOidShort (Of Key As OptimizedPersistable)  
    Inherits SortedSetBase (Of Key)
```

C++

```
[SerializableAttribute]  
generic<typename Key>  
where Key : OptimizedPersistable  
public ref class SortedSetOidShort : public SortedSetBase<Key>
```

F#

```
[<SerializableAttribute>]  
type SortedSetOidShort<'Key when 'Key : OptimizedPersistable> =  
    class  
        inherit SortedSetBase<'Key>  
    end
```




Type Parameters

Key




The type of elements in the set.

The SortedSetOidShort(Key) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------|---------------------------------------------------------------------------|
|  | SortedSetOidShort(Key)() | Constructs an OidShort based sorted set. |
|  | SortedSetOidShort(Key)(Int32) | Constructs an OidShort based sorted set with an initial presized capacity |
|  | SortedSetOidShort(Key)(VelocityDbComparer(Key)) | Constructs a new sorted set with a given object comparator. |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the object comparator (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal sorted list of objects contained in the set. (Overrides SortedSetBase(Key).Keys.) |
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |

Extension Methods




| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------|---------------------------------------------------------------------------|
|  | SortedSetOidShort(Key)() | Constructs an OidShort based sorted set. |
|  | SortedSetOidShort(Key)(Int32) | Constructs an OidShort based sorted set with an initial presized capacity |
|  | SortedSetOidShort(Key)(VelocityDbComparer(Key)) | Constructs a new sorted set with a given object comparator. |

See Also

[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key) Constructor

Constructs an OidShort based sorted set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetOidShort ()
```

VB

```
Public Sub New
```

C++

```
public:  
SortedSetOidShort ()
```

F#

```
new : unit -> SortedSetOidShort
```

See Also

[SortedSetOidShort\(Key\)Class](#)

[SortedSetOidShort\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key) Constructor (Int32)

Constructs an OidShort based sorted set with an initial presized capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetOidShort(  
    int capacity  
)
```

VB

```
Public Sub New (  
    capacity As Integer  
)
```

C++

```
public:  
SortedSetOidShort(  
    int capacity  
)
```

F#

```
new :  
    capacity : int -> SortedSetOidShort
```

Parameters

capacity

Type: [System.Int32](#)

See Also

[SortedSetOidShort\(Key\)Class](#)

[SortedSetOidShort\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key) Constructor (VelocityDbComparer(Key))

Constructs a new sorted set with a given object comparator.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSetOidShort (  
    VelocityDbComparer<Key> comparer  
)
```

VB

```
Public Sub New (  
    comparer As VelocityDbComparer(Of Key)  
)
```

C++

```
public:  
SortedSetOidShort (  
    VelocityDbComparer<Key>^ comparer  
)
```

F#

```
new :  
    comparer : VelocityDbComparer<'Key> -> SortedSetOidShort
```

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

See Also

[SortedSetOidShort\(Key\)Class](#)




[SortedSetOidShort\(Key\)Overload](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key).SortedSetOidShort(Key) Properties

The [SortedSetOidShort\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the object comparator (Overrides SortedSetBase(Key).Comparer.) |
|  | Keys | Gets the internal sorted list of objects contained in the set. (Overrides SortedSetBase(Key).Keys.) |
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |

See Also

[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key).Comparer Property

Gets the object comparator

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key).Keys Property

Gets the internal sorted list of objects contained in the set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Key> Keys { get; }
```

VB

```
Public Overrides ReadOnly Property Keys As IList(Of Key)  
    Get
```

C++

```
public:  
virtual property IList<Key>^ Keys {  
    IList<Key>^ get () override;  
}
```

F#

```
abstract Keys : IList<'Key> with get  
override Keys : IList<'Key> with get
```

Property Value

Type: [IList\(Key\)](#)

See Also

[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key).MaxNumberOfDatabases Property

[Missing <summary> documentation for

"P:VelocityDb.Collection.SortedSetOidShort`1.MaxNumberOfDatabases"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also



[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

SortedSetOidShort(Key).SortedSetOidShort(Key) Methods

The [SortedSetOidShort\(Key\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[SortedSetOidShort\(Key\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Class

A hash code based set

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Collection.VelocityDbHashSet(T)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class VelocityDbHashSet<T> : OptimizedPersistable,
    ICollection<T>, IEnumerable<T>, IEnumerable
```

VB

```
<SerializableAttribute>
Public Class VelocityDbHashSet(Of T)
    Inherits OptimizedPersistable
    Implements ICollection(Of T), IEnumerable(Of T),
    IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename T>
public ref class VelocityDbHashSet : public OptimizedPersistable,
    ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type VelocityDbHashSet<'T> =
    class
        inherit OptimizedPersistable
        interface ICollection<'T>
        interface IEnumerable<'T>
        interface IEnumerable
    end
```





Type Parameters

T




The object type of objects in the hashset

The VelocityDbHashSet(T) type exposes the following members.











Constructors









| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  VelocityDbHashSet(T)() | Creates a hashset using a default equality comparer |
|  VelocityDbHashSet(T)(IEnumerable(T)) | Creates a new hashset using the default equality comparer |
|  VelocityDbHashSet(T)(IEqualityComparer(T)) | Creates hashset with a given equality comparer |
|  VelocityDbHashSet(T)(IEnumerable(T), IEqualityComparer(T)) | Implementation Notes: Since resizes are relatively expensive (require rehashing), this attempts to minimize the need to resize by setting the initial capacity based on size of collection. |





Properties

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  Comparer | Gets the IEqualityComparer that is used to determine equality of keys for the VelocityDbHashSet. |
|  Count | Number of elements in this hashset |
|  ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage .) |



Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Add item to this VelocityDbHashSet. Returns bool indicating whether item was added (won't be added if already present) |
|  Clear | Remove all items from this set. This clears the elements but not the underlying buckets and slots array. Follow this call by TrimExcess to release these. |
|  Contains | Checks if this hashset contains the item |
|  CopyTo(T[]) | Copies the entire List to a compatible one-dimensional array, starting at the beginning of the target array. |
|  CopyTo(T[], Int32) | Copy items in this hashset to array, starting at arrayIndex |
|  CopyTo(T[], Int32, Int32) | Copies a range of elements from the List(Of T) to a compatible one-dimensional array, starting at the specified index of the target array. |
|  CreateSetComparer | Used for deep equality of VelocityDbHashSet testing |
|  ExceptWith | Remove items in other from this set. Modifies this set. |
|  GetEnumerator | Returns an enumerator that iterates through a collection |
|  GetObjectData | Serialization usage other than VelocityDb serialization |

| | |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IntersectWith | <p>Takes the intersection of this set with other. Modifies this set.</p> <p>Implementation Notes: We get better perf if other is a hashset using same equality comparer, because we get constant contains check in other. Resulting cost is $O(n1)$ to iterate over this. If we can't go above route, iterate over the other and mark intersection by checking contains in this. Then loop over and delete any unmarked elements. Total cost is $n2+n1$. Attempts to return early based on counts alone, using the property that the intersection of anything with the empty set is the empty set.</p> |
|  IsProperSubsetOf | <p>Checks if this is a proper subset of other (i.e. strictly contained in)</p> <p>Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a proper subset of a set that contains at least one element, but it's not a proper subset of the empty set. 2. If other has unique elements according to this equality comparer, and this has \geq the number of elements in other, then this can't be a proper subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check.</p> |
|  IsProperSupersetOf | <p>Checks if this is a proper superset of other (i.e. other strictly contained in this)</p> <p>Implementation Notes: This is slightly more complicated than above because we have to keep track if there was at least one element not contained in other. The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it can't be a proper superset of any set, even if other is the empty set. 2. If other is an empty set and this contains at least 1 element, then this is a proper superset. 3. If other has unique elements according to this equality comparer, and other's count is greater than or equal to this count, then this can't be a proper superset. Furthermore, if other has unique elements according to this equality comparer, we can use a faster element-wise check.</p> |
|  IsSubsetOf | <p>Checks if this is a subset of other.</p> <p>Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a subset of anything, including the empty set. 2. If other has unique elements according to this equality comparer, and this has more elements than other, then it can't be a subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check.</p> |
|  IsSupersetOf | <p>Checks if this is a superset of other</p> <p>Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If other has no elements (it's the empty set), then this is a superset, even if this is also the empty set. 2. If other has unique elements according to this equality comparer, and this has less than the number of elements in other, then this can't be a superset</p> |
|  Overlaps | <p>Checks if this set overlaps other (i.e. they share at least one item)</p> |
|  Remove | <p>Remove item from this hashset</p> |
|  RemoveWhere | <p>Remove elements that match specified predicate. Returns the number of elements removed</p> |

| | |
|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetEquals | Checks if this and other contain the same elements. This is set equality: duplicates and order are ignored |
|  SymmetricExceptWith | Takes symmetric difference (XOR) with other and this set. Modifies this set. |
|  TrimExcess | Sets the capacity of this list to the size of the list (rounded up to nearest prime), unless count is 0, in which case we release references. This method can be used to minimize a list's memory overhead once it is known that no new elements will be added to the list. To completely clear a list and release all memory referenced by the list, execute the following statements: list.Clear(); list.TrimExcess(); |
|  UnionWith | Take the union of this VelocityDbHashSet with other. Modifies this set. Implementation note: GetSuggestedCapacity (to increase capacity in advance avoiding multiple resizes ended up not being useful in practice; quickly gets to the point where it's a wasteful check. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Constructor

Overload List

| | Name | Description |
|---|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ⇒ | VelocityDbHashSet(T)() | Creates a hashset using a default equality comparer |
| ⇒ | VelocityDbHashSet(T)(IEnumerable(T)) | Creates a new hashset using the default equality comparer |
| ⇒ | VelocityDbHashSet(T)(IEqualityComparer(T)) | Creates hashset with a given equality comparer |
| ⇒ | VelocityDbHashSet(T)(IEnumerable(T), IEqualityComparer(T)) | Implementation Notes: Since resizes are relatively expensive (require rehashing), this attempts to minimize the need to resize by setting the initial capacity based on size of collection. |

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Constructor

Creates a hashset using a default equality comparer

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbHashSet ()
```

VB

```
Public Sub New
```

C++

```
public:  
VelocityDbHashSet ()
```

F#

```
new : unit -> VelocityDbHashSet
```

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDbHashSet\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Constructor (IEnumerable(T))

Creates a new hashset using the default equality comparer

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbHashSet (  
    IEnumerable<T> collection  
)
```

VB

```
Public Sub New (  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
VelocityDbHashSet (  
    IEnumerable<T>^ collection  
)
```

F#

```
new :  
    collection : IEnumerable<'T> -> VelocityDbHashSet
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

Initial objects added to the hashset

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDbHashSet\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Constructor (IEqualityComparer(T))

Creates hashset with a given equality comparer

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbHashSet (  
    IEqualityComparer<T> comparer  
)
```

VB

```
Public Sub New (  
    comparer As IEqualityComparer(Of T)  
)
```

C++

```
public:  
VelocityDbHashSet (  
    IEqualityComparer<T>^ comparer  
)
```

F#

```
new :  
    comparer : IEqualityComparer<'T> -> VelocityDbHashSet
```

Parameters

comparer

Type: [System.Collections.Generic.IEqualityComparer\(T\)](#)

[Missing <param name="comparer"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.#ctor(System.Collections.Generic.IEqualityComparer`0)"]

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDbHashSet\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T) Constructor (IEnumerable(T), IEqualityComparer(T))

Implementation Notes: Since resizes are relatively expensive (require rehashing), this attempts to minimize the need to resize by setting the initial capacity based on size of collection.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbHashSet (
    IEnumerable<T> collection,
    IEqualityComparer<T> comparer
)
```

VB

```
Public Sub New (
    collection As IEnumerable(Of T),
    comparer As IEqualityComparer(Of T)
)
```

C++

```
public:
VelocityDbHashSet (
    IEnumerable<T>^ collection,
    IEqualityComparer<T>^ comparer
)
```

F#

```
new :
    collection : IEnumerable<'T> *
    comparer : IEqualityComparer<'T> -> VelocityDbHashSet
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="collection"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.#ctor(System.Collections.Generic.IEnumerable{`0},System.Collections.Generic.IEqualityComparer{`0})"]

comparer

Type: [System.Collections.Generic.IEqualityComparer\(T\)](#)

[Missing <param name="comparer"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.#ctor(System.Collections.Generic.IEnumerable{`0},System.Collections.Generic.IEqualityComparer{`0})"]

VelocityDB Class Library

See Also

[VelocityDbHashSet\(T\)Class](#)




[VelocityDbHashSet\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).VelocityDbHashSet(T) Properties

The [VelocityDbHashSet\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the IEqualityComparer that is used to determine equality of keys for the VelocityDbHashSet. |
|  | Count | Number of elements in this hashset |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Comparer Property

Gets the IEqualityComparer that is used to determine equality of keys for the VelocityDbHashSet.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEqualityComparer<T> Comparer { get; }
```

VB

```
Public ReadOnly Property Comparer As IEqualityComparer(Of T)  
    Get
```

C++

```
public:  
property IEqualityComparer<T>^ Comparer {  
    IEqualityComparer<T>^ get ();  
}
```

F#

```
member Comparer : IEqualityComparer<'T> with get
```

Property Value

Type: [IEqualityComparer\(T\)](#)

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Count Property

Number of elements in this hashset

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

Implements

[ICollection\(T\).Count](#)

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).ObjectsPerPage Property

Limit to 1000 per page (instead of default 40000)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also














[VelocityDbHashSet\(T\)Class](#)










[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).VelocityDbHashSet(T) Methods


The [VelocityDbHashSet\(T\)](#) generic type exposes the following members.

Methods


| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Add item to this VelocityDbHashSet. Returns bool indicating whether item was added (won't be added if already present) |
|  Clear | Remove all items from this set. This clears the elements but not the underlying buckets and slots array. Follow this call by TrimExcess to release these. |
|  Contains | Checks if this hashset contains the item |
|  CopyTo(T[]) | Copies the entire List to a compatible one-dimensional array, starting at the beginning of the target array. |
|  CopyTo(T[], Int32) | Copy items in this hashset to array, starting at arrayIndex |
|  CopyTo(T[], Int32, Int32) | Copies a range of elements from the List(Of T) to a compatible one-dimensional array, starting at the specified index of the target array. |
|  CreateSetComparer | Used for deep equality of VelocityDbHashSet testing |
|  ExceptWith | Remove items in other from this set. Modifies this set. |
|  GetEnumerator | Returns an enumerator that iterates through a collection |
|  GetObjectData | Serialization usage other than VelocityDb serialization |
|  IntersectWith | Takes the intersection of this set with other. Modifies this set. Implementation Notes: We get better perf if other is a hashset using same equality comparer, because we get constant contains check in other. Resulting cost is O(n1) to iterate over this. If we can't go above route, iterate over the other and mark intersection by checking contains in this. Then loop over and delete any unmarked elements. Total cost is n2+n1. Attempts to return early based on counts alone, using the property that the intersection of anything with the empty set is the empty set. |
|  IsProperSubsetOf | Checks if this is a proper subset of other (i.e. strictly contained in) Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a proper subset of a set that contains at least one element, but it's not a proper subset of the empty set. 2. If other has unique elements according to this equality comparer, and this has >= the number of elements in other, then this can't be a proper subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check. |
|  IsProperSupersetOf | Checks if this is a proper superset of other (i.e. other strictly contained in this) Implementation Notes: This is slightly more complicated than above because we have to keep track if there was at least one element not contained in other. The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it can't be a proper |

| | | |
|-------------------------------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | superset of any set, even if other is the empty set. 2. If other is an empty set and this contains at least 1 element, then this is a proper superset. 3. If other has unique elements according to this equality comparer, and other's count is greater than or equal to this count, then this can't be a proper superset Furthermore, if other has unique elements according to this equality comparer, we can use a faster element-wise check. |
|  | IsSubsetOf | Checks if this is a subset of other. Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a subset of anything, including the empty set 2. If other has unique elements according to this equality comparer, and this has more elements than other, then it can't be a subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check. |
|  | IsSupersetOf | Checks if this is a superset of other Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If other has no elements (it's the empty set), then this is a superset, even if this is also the empty set. 2. If other has unique elements according to this equality comparer, and this has less than the number of elements in other, then this can't be a superset |
|  | Overlaps | Checks if this set overlaps other (i.e. they share at least one item) |
|  | Remove | Remove item from this hashset |
|  | RemoveWhere | Remove elements that match specified predicate. Returns the number of elements removed |
|  | SetEquals | Checks if this and other contain the same elements. This is set equality: duplicates and order are ignored |
|  | SymmetricExceptWith | Takes symmetric difference (XOR) with other and this set. Modifies this set. |
|  | TrimExcess | Sets the capacity of this list to the size of the list (rounded up to nearest prime), unless count is 0, in which case we release references. This method can be used to minimize a list's memory overhead once it is known that no new elements will be added to the list. To completely clear a list and release all memory referenced by the list, execute the following statements: list.Clear(); list.TrimExcess(); |
|  | UnionWith | Take the union of this VelocityDbHashSet with other. Modifies this set. Implementation note: GetSuggestedCapacity (to increase capacity in advance avoiding multiple resizes ended up not being useful in practice; quickly gets to the point where it's a wasteful check. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Add Method

Add item to this VelocityDbHashSet. Returns bool indicating whether item was added (won't be added if already present)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Add(  
    T item  
)
```

VB

```
Public Function Add (  
    item As T  
) As Boolean
```

C++

```
public:  
bool Add(  
    T item  
)
```

F#

```
member Add :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

[Missing <param name="item"/> documentation for "M:VelocityDb.Collection.VelocityDbHashSet`1.Add(`0)"]

Return Value

Type: [Boolean](#)

true if added, false if already present

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Clear Method

Remove all items from this set. This clears the elements but not the underlying buckets and slots array. Follow this call by TrimExcess to release these.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Clear()
```

VB

```
Public Sub Clear
```

C++

```
public:  
virtual void Clear() sealed
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Contains Method

Checks if this hashset contains the item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    T item  
)
```

VB

```
Public Function Contains (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    T item  
) sealed
```

F#

```
abstract Contains :  
    item : 'T -> bool  
override Contains :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

item to check for containment

Return Value

Type: [Boolean](#)

true if item contained; false if not

Implements

[ICollection\(T\).Contains\(T\)](#)




See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).CopyTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|  | CopyTo(T[]) | Copies the entire List to a compatible one-dimensional array, starting at the beginning of the target array. |
|  | CopyTo(T[], Int32) | Copy items in this hashset to array, starting at arrayIndex |
|  | CopyTo(T[], Int32, Int32) | Copies a range of elements from the List(Of T) to a compatible one-dimensional array, starting at the specified index of the target array. |

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).CopyTo Method (T[])

Copies the entire List to a compatible one-dimensional array, starting at the beginning of the target array.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo (
    T[] array
)
```

VB

```
Public Sub CopyTo (
    array As T()
)
```

C++

```
public:
void CopyTo (
    array<T>^ array
)
```

F#

```
member CopyTo :
    array : 'T[] -> unit
```

Parameters

array

Type: T[]

The one-dimensional Array that is the destination of the elements copied from List. The Array must have zero-based indexing.

See Also

[VelocityDbHashSet\(T\) Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).CopyTo Method (T[], Int32)

Copy items in this hashset to array, starting at arrayIndex

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    T[] array,  
    int arrayIndex  
)
```

VB

```
Public Sub CopyTo (  
    array As T(),  
    arrayIndex As Integer  
)
```

C++

```
public:  
virtual void CopyTo(  
    array<T>^ array,  
    int arrayIndex  
) sealed
```

F#

```
abstract CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit  
override CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit
```

Parameters

array

Type: *T*[]

array to add items to

arrayIndex

Type: [System.Int32](#)

index to start at

Implements

[ICollection\(T\).CopyTo\(T\[\], Int32\)](#)

VelocityDB Class Library

See Also

[VelocityDbHashSet\(T\)Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).CopyTo Method (T[], Int32, Int32)

Copies a range of elements from the List(Of T) to a compatible one-dimensional array, starting at the specified index of the target array.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    T[] array,  
    int arrayIndex,  
    int count  
)
```

VB

```
Public Sub CopyTo (  
    array As T(),  
    arrayIndex As Integer,  
    count As Integer  
)
```

C++

```
public:  
void CopyTo(  
    array<T>^ array,  
    int arrayIndex,  
    int count  
)
```

F#

```
member CopyTo :  
    array : 'T[] *  
    arrayIndex : int *  
    count : int -> unit
```

Parameters

array

Type: *T*[]

The one-dimensional Array that is the destination of the elements copied from List(Of T). The Array must have zero-based indexing.

arrayIndex

Type: [System.Int32](#)

The zero-based index in the source List(Of T) at which copying begins.

count

VelocityDB Class Library

Type: [System.Int32](#)

The number of elements to copy.

See Also

[VelocityDbHashSet\(T\)Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).CreateSetComparer Method

Used for deep equality of VelocityDbHashSet testing

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEqualityComparer<VelocityDbHashSet<T>> CreateSetComparer ()
```

VB

```
Public Shared Function CreateSetComparer As IEqualityComparer (Of  
VelocityDbHashSet (Of T))
```

C++

```
public:  
static IEqualityComparer<VelocityDbHashSet<T>^>^ CreateSetComparer ()
```

F#

```
static member CreateSetComparer : unit ->  
IEqualityComparer<VelocityDbHashSet<'T>>
```

Return Value

Type: [IEqualityComparer\(VelocityDbHashSet\(T\)\)](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.CreateSetComparer"]

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).ExceptWith Method

Remove items in other from this set. Modifies this set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExceptWith(  
    IEnumerable<T> other  
)
```

VB

```
Public Sub ExceptWith (  
    other As IEnumerable(Of T)  
)
```

C++

```
public:  
void ExceptWith(  
    IEnumerable<T>^ other  
)
```

F#

```
member ExceptWith :  
    other : IEnumerable<'T> -> unit
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

enumerable with items to remove

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).GetEnumerator Method

Returns an enumerator that iterates through a collection

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbHashSet<T>.Enumerator GetEnumerator()
```

VB

```
Public Function GetEnumerator As VelocityDbHashSet<T>.Enumerator
```

C++

```
public:  
VelocityDbHashSet<T>.Enumerator GetEnumerator()
```

F#

```
member GetEnumerator : unit -> VelocityDbHashSet<T>.Enumerator
```

Return Value

Type: [VelocityDbHashSet\(T\).Enumerator](#)

An Enumerator object that can be used to iterate through the collection.

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).GetObjectData Method

Serialization usage other than VelocityDb serialization

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void GetObjectData(  
    SerializationInfo info,  
    StreamingContext context  
)
```

VB

```
Public Overridable Sub GetObjectData (  
    info As SerializationInfo,  
    context As StreamingContext  
)
```

C++

```
public:  
virtual void GetObjectData(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

F#

```
abstract GetObjectData :  
    info : SerializationInfo *  
    context : StreamingContext -> unit  
override GetObjectData :  
    info : SerializationInfo *  
    context : StreamingContext -> unit
```

Parameters

info

Type: [System.Runtime.Serialization.SerializationInfo](#)

[Missing <param name="info"/> documentation for

**"M:VelocityDb.Collection.VelocityDbHashSet`1.GetObjectData(System.Runtime.Serialization.Serializa
tionInfo,System.Runtime.Serialization.StreamingContext)"]**

context

Type: [System.Runtime.Serialization.StreamingContext](#)

[Missing <param name="context"/> documentation for

**"M:VelocityDb.Collection.VelocityDbHashSet`1.GetObjectData(System.Runtime.Serialization.Serializa
tionInfo,System.Runtime.Serialization.StreamingContext)"]**

VelocityDB Class Library

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).IntersectWith Method

Takes the intersection of this set with other. Modifies this set. Implementation Notes: We get better perf if other is a hashset using same equality comparer, because we get constant contains check in other. Resulting cost is $O(n1)$ to iterate over this. If we can't go above route, iterate over the other and mark intersection by checking contains in this. Then loop over and delete any unmarked elements. Total cost is $n2+n1$. Attempts to return early based on counts alone, using the property that the intersection of anything with the empty set is the empty set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void IntersectWith(  
    IEnumerable<T> other  
)
```

VB

```
Public Sub IntersectWith (  
    other As IEnumerable(Of T)  
)
```

C++

```
public:  
void IntersectWith(  
    IEnumerable<T>^ other  
)
```

F#

```
member IntersectWith :  
    other : IEnumerable<'T> -> unit
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

enumerable with items to add

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).IsProperSubsetOf Method

Checks if this is a proper subset of other (i.e. strictly contained in) Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a proper subset of a set that contains at least one element, but it's not a proper subset of the empty set. 2. If other has unique elements according to this equality comparer, and this has \geq the number of elements in other, then this can't be a proper subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsProperSubsetOf(
    IEnumerable<T> other
)
```

VB

```
Public Function IsProperSubsetOf (
    other As IEnumerable(Of T)
) As Boolean
```

C++

```
public:
bool IsProperSubsetOf(
    IEnumerable<T>^ other
)
```

F#

```
member IsProperSubsetOf :
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.IsProperSubsetOf(System.Collections.Generic.IEnumerable{`0})"]

Return Value

Type: [Boolean](#)

true if this is a proper subset of other; false if not

See Also

[VelocityDbHashSet\(T\)Class](#)

VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).IsProperSupersetOf Method

Checks if this is a proper superset of other (i.e. other strictly contained in this) Implementation Notes: This is slightly more complicated than above because we have to keep track if there was at least one element not contained in other. The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it can't be a proper superset of any set, even if other is the empty set. 2. If other is an empty set and this contains at least 1 element, then this is a proper superset. 3. If other has unique elements according to this equality comparer, and other's count is greater than or equal to this count, then this can't be a proper superset Furthermore, if other has unique elements according to this equality comparer, we can use a faster element-wise check.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsProperSupersetOf (
    IEnumerable<T> other
)
```

VB

```
Public Function IsProperSupersetOf (
    other As IEnumerable(Of T)
) As Boolean
```

C++

```
public:
bool IsProperSupersetOf (
    IEnumerable<T>^ other
)
```

F#

```
member IsProperSupersetOf :
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for "M:VelocityDb.Collection.VelocityDbHashSet`1.IsProperSupersetOf(System.Collections.Generic.IEnumerable{`0})"]

Return Value

Type: [Boolean](#)

true if this is a proper superset of other; false if not

VelocityDB Class Library

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).IsSubsetOf Method

Checks if this is a subset of other. Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If this is the empty set, then it's a subset of anything, including the empty set 2. If other has unique elements according to this equality comparer, and this has more elements than other, then it can't be a subset. Furthermore, if other is a hashset using the same equality comparer, we can use a faster element-wise check.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsSubsetOf(
    IEnumerable<T> other
)
```

VB

```
Public Function IsSubsetOf (
    other As IEnumerable(Of T)
) As Boolean
```

C++

```
public:
bool IsSubsetOf(
    IEnumerable<T>^ other
)
```

F#

```
member IsSubsetOf :
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.IsSubsetOf(System.Collections.Generic.IEnumerable`0)"]

Return Value

Type: [Boolean](#)

true if this is a subset of other; false if not

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).IsSupersetOf Method

Checks if this is a superset of other
Implementation Notes: The following properties are used up-front to avoid element-wise checks: 1. If other has no elements (it's the empty set), then this is a superset, even if this is also the empty set. 2. If other has unique elements according to this equality comparer, and this has less than the number of elements in other, then this can't be a superset

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsSupersetOf(  
    IEnumerable<T> other  
)
```

VB

```
Public Function IsSupersetOf (  
    other As IEnumerable(Of T)  
) As Boolean
```

C++

```
public:  
bool IsSupersetOf(  
    IEnumerable<T>^ other  
)
```

F#

```
member IsSupersetOf :  
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.IsSupersetOf(System.Collections.Generic.IEnumerable<T>)"

Return Value

Type: [Boolean](#)

true if this is a superset of other; false if not

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Overlaps Method

Checks if this set overlaps other (i.e. they share at least one item)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Overlaps(  
    IEnumerable<T> other  
)
```

VB

```
Public Function Overlaps (  
    other As IEnumerable(Of T)  
) As Boolean
```

C++

```
public:  
bool Overlaps(  
    IEnumerable<T>^ other  
)
```

F#

```
member Overlaps :  
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.Overlaps(System.Collections.Generic.IEnumerable{`0})"
"]

Return Value

Type: [Boolean](#)

true if these have at least one common element; false if disjoint

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Remove Method

Remove item from this hashset

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Remove(  
    T item  
)
```

VB

```
Public Function Remove (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    T item  
) sealed
```

F#

```
abstract Remove :  
    item : 'T -> bool  
override Remove :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

item to remove

Return Value

Type: [Boolean](#)

true if removed; false if not (i.e. if the item wasn't in the VelocityDbHashSet)

Implements

[ICollection\(T\).Remove\(T\)](#)

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).RemoveWhere Method

Remove elements that match specified predicate. Returns the number of elements removed

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int RemoveWhere(  
    Predicate<T> match  
)
```

VB

```
Public Function RemoveWhere (  
    match As Predicate(Of T)  
) As Integer
```

C++

```
public:  
int RemoveWhere(  
    Predicate<T>^ match  
)
```

F#

```
member RemoveWhere :  
    match : Predicate<'T> -> int
```

Parameters

match

Type: [System.Predicate\(T\)](#)

[Missing <param name="match"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.RemoveWhere(System.Predicate`0)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.RemoveWhere(System.Predicate`0)"]

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).SetEquals Method

Checks if this and other contain the same elements. This is set equality: duplicates and order are ignored

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SetEquals(  
    IEnumerable<T> other  
)
```

VB

```
Public Function SetEquals (  
    other As IEnumerable(Of T)  
) As Boolean
```

C++

```
public:  
bool SetEquals(  
    IEnumerable<T>^ other  
)
```

F#

```
member SetEquals :  
    other : IEnumerable<'T> -> bool
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="other"/> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.SetEquals(System.Collections.Generic.IEnumerable{`0})"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.VelocityDbHashSet`1.SetEquals(System.Collections.Generic.IEnumerable{`0})"]

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).SymmetricExceptWith Method

Takes symmetric difference (XOR) with other and this set. Modifies this set.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SymmetricExceptWith(  
    IEnumerable<T> other  
)
```

VB

```
Public Sub SymmetricExceptWith (  
    other As IEnumerable(Of T)  
)
```

C++

```
public:  
void SymmetricExceptWith(  
    IEnumerable<T>^ other  
)
```

F#

```
member SymmetricExceptWith :  
    other : IEnumerable<'T> -> unit
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

enumerable with items to XOR

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).TrimExcess Method

Sets the capacity of this list to the size of the list (rounded up to nearest prime), unless count is 0, in which case we release references. This method can be used to minimize a list's memory overhead once it is known that no new elements will be added to the list. To completely clear a list and release all memory referenced by the list, execute the following statements: list.Clear(); list.TrimExcess();

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void TrimExcess ()
```

VB

```
Public Sub TrimExcess
```

C++

```
public:  
void TrimExcess ()
```

F#

```
member TrimExcess : unit -> unit
```

See Also

[VelocityDbHashSet\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).UnionWith Method

Take the union of this VelocityDbHashSet with other. Modifies this set. Implementation note: GetSuggestedCapacity (to increase capacity in advance avoiding multiple resizes ended up not being useful in practice; quickly gets to the point where it's a wasteful check.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void UnionWith(  
    IEnumerable<T> other  
)
```

VB

```
Public Sub UnionWith (  
    other As IEnumerable(Of T)  
)
```

C++

```
public:  
void UnionWith(  
    IEnumerable<T>^ other  
)
```

F#

```
member UnionWith :  
    other : IEnumerable<'T> -> unit
```

Parameters

other

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

enumerable with items to add

See Also

[VelocityDbHashSet\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Enumerator Structure

Enumerates the elements of a VelocityDbHashSet(Of T).

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
[HostProtectionAttribute(SecurityAction.LinkDemand, MayLeakOnAbort = true)]
public struct Enumerator : IEnumerator<T>,
    IDisposable, IEnumerator
```

VB

```
<SerializableAttribute>
<HostProtectionAttribute(SecurityAction.LinkDemand, MayLeakOnAbort := true)>
Public Structure Enumerator
    Implements IEnumerator(Of T), IDisposable, IEnumerator
```

C++


```
[SerializableAttribute]
[HostProtectionAttribute(SecurityAction::LinkDemand, MayLeakOnAbort = true)]
public value class Enumerator : IEnumerator<T>,
    IDisposable, IEnumerator
```

F#


```
[<SealedAttribute>]
[<SerializableAttribute>]
[<HostProtectionAttribute(SecurityAction.LinkDemand, MayLeakOnAbort = true)>]
type Enumerator =
    struct
        interface IEnumerator<'T>
        interface IDisposable
        interface IEnumerator
    end
```

The VelocityDbHashSet(T).Enumerator generic type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|---------------------------------|
|  | Current | The iterator current Key object |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|----------------------------------------------------------------------------------------------------------|
|  | Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |

| | |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  | MoveNext Advances the enumerator to the next element of the collection. |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|


See Also

[VelocityDb.Collection Namespace](#)

Enumerator.Enumerator Properties

The [VelocityDbHashSet\(T\).Enumerator](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|---------------------------------|
|  | Current | The iterator current Key object |

See Also

[VelocityDbHashSet\(T\).Enumerator Structure](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Enumerator.Current Property

The iterator current Key object

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T Current { get; }
```

VB

```
Public ReadOnly Property Current As T  
    Get
```

C++

```
public:  
virtual property T Current {  
    T get () sealed;  
}
```

F#

```
abstract Current : 'T with get  
override Current : 'T with get
```

Property Value

Type: *T*

Implements

[IEnumerator\(T\).Current](#)

See Also



[VelocityDbHashSet\(T\).Enumerator Structure](#)

[VelocityDb.Collection Namespace](#)

Enumerator.Enumerator Methods

The [VelocityDbHashSet\(T\).Enumerator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------|
|  | Dispose | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources. |
|  | MoveNext | Advances the enumerator to the next element of the collection. |

See Also

[VelocityDbHashSet\(T\).Enumerator Structure](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Enumerator.Dispose Method

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[VelocityDbHashSet\(T\).Enumerator Structure](#)

[VelocityDb.Collection Namespace](#)

VelocityDbHashSet(T).Enumerator.MoveNext Method

Advances the enumerator to the next element of the collection.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool MoveNext ()
```

VB

```
Public Function MoveNext As Boolean
```

C++

```
public:  
virtual bool MoveNext () sealed
```

F#

```
abstract MoveNext : unit -> bool  
override MoveNext : unit -> bool
```

Return Value

Type: [Boolean](#)

true if the enumerator was successfully advanced to the next element; false if the enumerator has passed the end of the collection.

Implements

[IEnumerator.MoveNext\(\)](#)

See Also

[VelocityDbHashSet\(T\).Enumerator Structure](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T) Class

A list with an Oid containing objects ordered by index.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.VelocityDbList\(T\)](#)

[VelocityDb.Collection.VelocityDbListOidShort\(T\)](#)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class VelocityDbList<T> : OptimizedPersistable,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

VB

```
<SerializableAttribute>
Public Class VelocityDbList(Of T)
    Inherits OptimizedPersistable
    Implements IList(Of T), ICollection(Of T),
    IEnumerable(Of T), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename T>
public ref class VelocityDbList : public OptimizedPersistable,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type VelocityDbList<'T> =
    class
        inherit OptimizedPersistable
        interface IList<'T>
        interface ICollection<'T>
        interface IEnumerable<'T>
        interface IEnumerable
    end
```



Type Parameters

T








The type of objects contained in this kind of list

The VelocityDbList(T) type exposes the following members.















Constructors





| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|------------------------------------------------|
|  | VelocityDbList(T)() | Constructs an empty list |
|  | VelocityDbList(T)(Int32) | Constructs a new list with a presized capacity |

Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list |
|  | IsFixedSize | Always false |
|  | IsReadOnly | Always false |
|  | IsSynchronized | Always false |
|  | Item | Gets or sets an item at a certain index |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | SyncRoot | Used for thread safety |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
|  | Add(T) | Adds an item to a list |
|  | Add(T, Int32) | Adds an item to a list and if array needs to grow to hold new item, it grows to a given max size. |
|  | AddRange | Adds the elements of the given collection to the end of this list. |
|  | BinarySearch | Assuming list elements are sorted by default sort order, finds the position of a an element or position where it would belong |
|  | Clear | Removes all items from the list and frees the array |
|  | Contains | Checks if an item is contained in the list |
|  | CopyTo(Array, Int32) | Copies list to an array |
|  | CopyTo(T[], Int32) | Copies list items to an array |
|  | GetEnumerator | |
|  | IndexOf | Finds the first index of an item |
|  | Insert(Int32, T) | Inserts an item at a specified index |
|  | Insert(Int32, T, Int32) | Inserts an item at a specified index |
|  | InsertRange | Inserts the elements of the given collection at a given index. |
|  | Last | Get the value at the last position in the list |

| | |
|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  PersistMyReferences | (Overrides OptimizedPersistable.PersistMyReferences(SessionBase, Boolean).) |
|  Remove | Remove an item |
|  RemoveAt | Removes an item at a specified index |
|  RemoveRange | Removes a range of items |

Extension Methods



| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Get(T) | (Defined by GraphHelpers.) |
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

VelocityDbList(T) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|------------------------------------------------|
|  | VelocityDbList(T)() | Constructs an empty list |
|  | VelocityDbList(T)(Int32) | Constructs a new list with a presized capacity |

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T) Constructor

Constructs an empty list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbList ()
```

VB

```
Public Sub New
```

C++

```
public:  
VelocityDbList ()
```

F#

```
new : unit -> VelocityDbList
```

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDbList\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T) Constructor (Int32)

Constructs a new list with a presized capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbList(  
    int capacity  
)
```

VB

```
Public Sub New (  
    capacity As Integer  
)
```

C++

```
public:  
VelocityDbList(  
    int capacity  
)
```

F#

```
new :  
    capacity : int -> VelocityDbList
```

Parameters

capacity

Type: [System.Int32](#)

[Missing <param name="capacity"/> documentation for "M:VelocityDb.Collection.VelocityDbList`1.#ctor(System.Int32)"]

See Also

[VelocityDbList\(T\)Class](#)








[VelocityDbList\(T\)Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).VelocityDbList(T) Properties

The [VelocityDbList\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list |
|  | IsFixedSize | Always false |
|  | IsReadOnly | Always false |
|  | IsSynchronized | Always false |
|  | Item | Gets or sets an item at a certain index |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | SyncRoot | Used for thread safety |

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Count Property

Number of items in list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

Implements

[ICollection\(T\).Count](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).IsFixedSize Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsFixedSize { get; }
```

VB

```
Public ReadOnly Property IsFixedSize As Boolean  
    Get
```

C++

```
public:  
property bool IsFixedSize {  
    bool get ();  
}
```

F#

```
member IsFixedSize : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).IsReadOnly Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsReadOnly { get; }
```

VB

```
Public ReadOnly Property IsReadOnly As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsReadOnly {  
    bool get () sealed;  
}
```

F#

```
abstract IsReadOnly : bool with get  
override IsReadOnly : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[ICollection\(T\).IsReadOnly](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).IsSynchronized Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsSynchronized { get; }
```

VB

```
Public ReadOnly Property IsSynchronized As Boolean  
    Get
```

C++

```
public:  
property bool IsSynchronized {  
    bool get ();  
}
```

F#

```
member IsSynchronized : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Item Property

Gets or sets an item at a certain index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual T this[
    int index
] { get; set; }
```

VB

```
Public Overridable Default Property Item (
    index As Integer
) As T
    Get
    Set
```

C++

```
public:
virtual property T default[int index] {
    T get (int index);
    void set (int index, T value);
}
```

F#

```
abstract Item : 'T with get, set
override Item : 'T with get, set
```

Parameters

index

Type: [System.Int32](#)

the item index

Return Value

Type: *T*

the item at the specified index

Implements

[IList\(T\).Item\(Int32\)](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).ObjectsPerPage Property

Limit to 1000 per page (instead of default 40000)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).SyncRoot Property

Used for thread safety

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object SyncRoot { get; }
```

VB

```
Public ReadOnly Property SyncRoot As Object  
    Get
```

C++

```
public:  
property Object^ SyncRoot {  
    Object^ get ();  
}
```

F#

```
member SyncRoot : Object with get
```

Property Value

Type: [Object](#)

See Also



















[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)




VelocityDbList(T).VelocityDbList(T) Methods

The [VelocityDbList\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  Add(T) | Adds an item to a list |
|  Add(T, Int32) | Adds an item to a list and if array needs to grow to hold new item, it grows to a given max size. |
|  AddRange | Adds the elements of the given collection to the end of this list. |
|  BinarySearch | Assuming list elements are sorted by default sort order, finds the position of an element or position where it would belong |
|  Clear | Removes all items from the list and frees the array |
|  Contains | Checks if an item is contained in the list |
|  CopyTo(Array, Int32) | Copies list to an array |
|  CopyTo(T[], Int32) | Copies list items to an array |
|  GetEnumerator | |
|  IndexOf | Finds the first index of an item |
|  Insert(Int32, T) | Inserts an item at a specified index |
|  Insert(Int32, T, Int32) | Inserts an item at a specified index |
|  InsertRange | Inserts the elements of the given collection at a given index. |
|  Last | Get the value at the last position in the list |
|  PersistMyReferences | (Overrides OptimizedPersistable.PersistMyReferences(SessionBase, Boolean) .) |
|  Remove | Remove an item |
|  RemoveAt | Removes an item at a specified index |
|  RemoveRange | Removes a range of items |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Get(T) | (Defined by GraphHelpers .) |
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also



[VelocityDbList\(T\)Class](#)

VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Add Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------------------|
|  | Add(T) | Adds an item to a list |
|  | Add(T, Int32) | Adds an item to a list and if array needs to grow to hold new item, it grows to a given max size. |

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Add Method (T)

Adds an item to a list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Add(  
    T item  
)
```

VB

```
Public Overridable Sub Add (  
    item As T  
)
```

C++

```
public:  
virtual void Add(  
    T item  
)
```

F#

```
abstract Add :  
    item : 'T -> unit  
override Add :  
    item : 'T -> unit
```

Parameters

item

Type: *T*

item being added

Implements

[ICollection\(T\).Add\(T\)](#)

See Also

[VelocityDbList\(T\) Class](#)

[Add Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Add Method (T, Int32)

Adds an item to a list and if array needs to grow to hold new item, it grows to a given max size.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Add(  
    T item,  
    int maxArraySize  
)
```

VB

```
Public Overridable Sub Add (  
    item As T,  
    maxArraySize As Integer  
)
```

C++

```
public:  
virtual void Add(  
    T item,  
    int maxArraySize  
)
```

F#

```
abstract Add :  
    item : 'T *  
    maxArraySize : int -> unit  
override Add :  
    item : 'T *  
    maxArraySize : int -> unit
```

Parameters

item

Type: *T*

item being added

maxArraySize

Type: [System.Int32](#)

the maximum size to use as internal array size

See Also

[VelocityDbList\(T\)Class](#)

[Add Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).AddRange Method

Adds the elements of the given collection to the end of this list.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddRange (  
    IEnumerable<T> collection  
)
```

VB

```
Public Sub AddRange (  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
void AddRange (  
    IEnumerable<T>^ collection  
)
```

F#

```
member AddRange :  
    collection : IEnumerable<'T> -> unit
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be added

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).BinarySearch Method

Assuming list elements are sorted by default sort order, finds the position of a an element or position where it would belong

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int BinarySearch(  
    T aKey,  
    out bool isEqual  
)
```

VB

```
Public Function BinarySearch (  
    aKey As T,  
    <OutAttribute> ByRef isEqual As Boolean  
) As Integer
```

C++

```
public:  
int BinarySearch(  
    T aKey,  
    [OutAttribute] bool% isEqual  
)
```

F#

```
member BinarySearch :  
    aKey : 'T *  
    isEqual : bool byref -> int
```

Parameters

aKey

Type: *T*

Element to look for

isEqual

Type: [System.Boolean](#)

true if match found; otherwise false

Return Value

Type: [Int32](#)

Index position

VelocityDB Class Library

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Clear Method

Removes all items from the list and frees the array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Clear ()
```

VB

```
Public Overridable Sub Clear
```

C++

```
public:  
virtual void Clear ()
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Contains Method

Checks if an item is contained in the list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    T item  
)
```

VB

```
Public Function Contains (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    T item  
) sealed
```

F#

```
abstract Contains :  
    item : 'T -> bool  
override Contains :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to look for

Return Value

Type: [Boolean](#)

true if item was found; otherwise false

Implements

[ICollection\(T\).Contains\(T\)](#)



See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).CopyTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------|
|  | CopyTo(Array, Int32) | Copies list to an array |
|  | CopyTo(T[], Int32) | Copies list items to an array |

See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).CopyTo Method (Array, Int32)

Copies list to an array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    Array array,  
    int arrayIndex  
)
```

VB

```
Public Sub CopyTo (  
    array As Array,  
    arrayIndex As Integer  
)
```

C++

```
public:  
void CopyTo(  
    Array^ array,  
    int arrayIndex  
)
```

F#

```
member CopyTo :  
    array : Array *  
    arrayIndex : int -> unit
```

Parameters

array

Type: [System.Array](#)

the array to copy to

arrayIndex

Type: [System.Int32](#)

start index in array for copy

See Also

[VelocityDbList\(T\)Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).CopyTo Method (T[], Int32)

Copies list items to an array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    T[] array,  
    int arrayIndex  
)
```

VB

```
Public Sub CopyTo (  
    array As T(),  
    arrayIndex As Integer  
)
```

C++

```
public:  
virtual void CopyTo(  
    array<T>^ array,  
    int arrayIndex  
) sealed
```

F#

```
abstract CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit  
override CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit
```

Parameters

array

Type: *T*[]

the array to copy to

arrayIndex

Type: [System.Int32](#)

start array index of copy

Implements

[ICollection\(T\).CopyTo\(T\[\], Int32\)](#)

VelocityDB Class Library

See Also

[VelocityDbList\(T\)Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).GetEnumerator Method

[Missing <summary> documentation for "M:VelocityDb.Collection.VelocityDbList`1.GetEnumerator"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<T> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of T)
```

C++

```
public:  
virtual IEnumerable<T>^ GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<'T>  
override GetEnumerator : unit -> IEnumerable<'T>
```

Return Value

Type: [IEnumerable\(T\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.VelocityDbList`1.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).IndexOf Method

Finds the first index of an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int IndexOf(  
    T item  
)
```

VB

```
Public Function IndexOf (  
    item As T  
) As Integer
```

C++

```
public:  
virtual int IndexOf(  
    T item  
) sealed
```

F#

```
abstract IndexOf :  
    item : 'T -> int  
override IndexOf :  
    item : 'T -> int
```

Parameters

item

Type: *T*

item to look for

Return Value

Type: [Int32](#)

-1 if item was not found or first index of item

Implements

[IList\(T\).IndexOf\(T\)](#)



See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Insert Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------|
|  | Insert(Int32, T) | Inserts an item at a specified index |
|  | Insert(Int32, T, Int32) | Inserts an item at a specified index |

See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Insert Method (Int32, T)

Inserts an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Insert(  
    int index,  
    T item  
)
```

VB

```
Public Overridable Sub Insert (  
    index As Integer,  
    item As T  
)
```

C++

```
public:  
virtual void Insert(  
    int index,  
    T item  
)
```

F#

```
abstract Insert :  
    index : int *  
    item : 'T -> unit  
override Insert :  
    index : int *  
    item : 'T -> unit
```

Parameters

index

Type: [System.Int32](#)

insert at this index

item

Type: *T*

the item being inserted

Implements

[IList\(T\).Insert\(Int32, T\)](#)

VelocityDB Class Library

See Also

[VelocityDbList\(T\)Class](#)

[Insert Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Insert Method (Int32, T, Int32)

Inserts an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Insert(  
    int index,  
    T item,  
    int maxArraySize  
)
```

VB

```
Public Overridable Sub Insert (  
    index As Integer,  
    item As T,  
    maxArraySize As Integer  
)
```

C++

```
public:  
virtual void Insert(  
    int index,  
    T item,  
    int maxArraySize  
)
```

F#

```
abstract Insert :  
    index : int *  
    item : 'T *  
    maxArraySize : int -> unit  
override Insert :  
    index : int *  
    item : 'T *  
    maxArraySize : int -> unit
```

Parameters

index

Type: [System.Int32](#)

insert at this index

item

Type: *T*

the item being inserted

VelocityDB Class Library

maxArraySize

Type: [System.Int32](#)

Restrict array growth to this maximum size (if array needs to grow)

See Also

[VelocityDbList\(T\)Class](#)

[Insert Overload](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).InsertRange Method

Inserts the elements of the given collection at a given index.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InsertRange(  
    int index,  
    IEnumerable<T> collection  
)
```

VB

```
Public Sub InsertRange (  
    index As Integer,  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
void InsertRange(  
    int index,  
    IEnumerable<T>^ collection  
)
```

F#

```
member InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit
```

Parameters

index

Type: [System.Int32](#)

start insert at this index

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be inserted

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Last Method

Get the value at the last position in the list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T Last()
```

VB

```
Public Function Last As T
```

C++

```
public:  
T Last()
```

F#

```
member Last : unit -> 'T
```

Return Value

Type: *T*

The value at the last position in the list

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).PersistMyReferences Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.VelocityDbList`1.PersistMyReferences(VelocityDb.Session.SessionBase,System.Boolean)"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void PersistMyReferences (
    SessionBase session,
    bool inFlush
)
```

VB

```
Public Overrides Sub PersistMyReferences (
    session As SessionBase,
    inFlush As Boolean
)
```

C++

```
public:
virtual void PersistMyReferences (
    SessionBase^ session,
    bool inFlush
) override
```

F#

```
abstract PersistMyReferences :
    session : SessionBase *
    inFlush : bool -> unit
override PersistMyReferences :
    session : SessionBase *
    inFlush : bool -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.VelocityDbList`1.PersistMyReferences(VelocityDb.Session.SessionBase,System.Boolean)"]

inFlush

Type: [System.Boolean](#)

[Missing <param name="inFlush"/> documentation for "M:VelocityDb.Collection.VelocityDbList`1.PersistMyReferences(VelocityDb.Session.SessionBase,System.Boolean)"]

Implements

[IOptimizedPersistable.PersistMyReferences\(SessionBase, Boolean\)](#)

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).Remove Method

Remove an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool Remove(  
    T item  
)
```

VB

```
Public Overridable Function Remove (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Remove(  
    T item  
)
```

F#

```
abstract Remove :  
    item : 'T -> bool  
override Remove :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to remove

Return Value

Type: [Boolean](#)

true if item was found and removed; otherwise false

Implements

[ICollection\(T\).Remove\(T\)](#)

See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).RemoveAt Method

Removes an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RemoveAt (  
    int index  
)
```

VB

```
Public Overridable Sub RemoveAt (  
    index As Integer  
)
```

C++

```
public:  
virtual void RemoveAt (  
    int index  
)
```

F#

```
abstract RemoveAt :  
    index : int -> unit  
override RemoveAt :  
    index : int -> unit
```

Parameters

index

Type: [System.Int32](#)

the index of the item to remove

Implements

[IList\(T\).RemoveAt\(Int32\)](#)

See Also

[VelocityDbList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbList(T).RemoveRange Method

Removes a range of items

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RemoveRange (  
    int index,  
    int howMany  
)
```

VB

```
Public Overridable Sub RemoveRange (  
    index As Integer,  
    howMany As Integer  
)
```

C++

```
public:  
virtual void RemoveRange (  
    int index,  
    int howMany  
)
```

F#

```
abstract RemoveRange :  
    index : int *  
    howMany : int -> unit  
override RemoveRange :  
    index : int *  
    howMany : int -> unit
```

Parameters

index

Type: [System.Int32](#)

start index

howMany

Type: [System.Int32](#)

how many to remove

See Also

[VelocityDbList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbListOidShort(T) Class

List of objects where list and all elements of a list are contained within a single [Database](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.VelocityDbList\(T\)](#)

VelocityDb.Collection.VelocityDbListOidShort(T)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VelocityDbListOidShort<T> : VelocityDbList<T>,  
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

VB

```
<SerializableAttribute>  
Public Class VelocityDbListOidShort(Of T)  
    Inherits VelocityDbList(Of T)  
    Implements IList(Of T), ICollection(Of T),  
    IEnumerable(Of T), IEnumerable
```

C++

```
[SerializableAttribute]  
generic<typename T>  
public ref class VelocityDbListOidShort : public VelocityDbList<T>,  
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<SerializableAttribute>]  
type VelocityDbListOidShort<'T> =  
    class  
        inherit VelocityDbList<'T>  
        interface IList<'T>  
        interface ICollection<'T>  
        interface IEnumerable<'T>  
        interface IEnumerable  
    end
```


Type Parameters

T


The element type of a list.

The VelocityDbListOidShort(T) type exposes the following members.




Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|---------------------------------------------|
|  | VelocityDbListOidShort(T) | Creates a new list with a presized capacity |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------|
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | Get(T) | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection Namespace](#)

VelocityDbListOidShort(T) Constructor

Creates a new list with a presized capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbListOidShort(  
    int capacity  
)
```

VB

```
Public Sub New (  
    capacity As Integer  
)
```

C++

```
public:  
VelocityDbListOidShort(  
    int capacity  
)
```

F#

```
new :  
    capacity : int -> VelocityDbListOidShort
```

Parameters

capacity

Type: [System.Int32](#)

See Also


[VelocityDbListOidShort\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbListOidShort(T).VelocityDbListOidShort(T) Properties

The [VelocityDbListOidShort\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------------------|
|  | MaxNumberOfDatabases | (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |

See Also

[VelocityDbListOidShort\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbListOidShort(T).MaxNumberOfDatabases Property

[Missing <summary> documentation for

"P:VelocityDb.Collection.VelocityDbListOidShort`1.MaxNumberOfDatabases"]

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also




[VelocityDbListOidShort\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

VelocityDbListOidShort(T).VelocityDbListOidShort(T) Methods

The [VelocityDbListOidShort\(T\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | Get(T) | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDbListOidShort\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T) Class

List with an Id containing object Ids ordered by index.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.WeakReferenceListBase\(T\)](#)

VelocityDb.Collection.WeakReferenceList(T)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class WeakReferenceList<T> : WeakReferenceListBase<T>,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class WeakReferenceList(Of T As IOptimizedPersistable)
    Inherits WeakReferenceListBase(Of T)
    Implements IList(Of T), ICollection(Of T),
        IEnumerable(Of T), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename T>
where T : IOptimizedPersistable
public ref class WeakReferenceList : public WeakReferenceListBase<T>,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type WeakReferenceList<'T when 'T : IOptimizedPersistable> =
    class
        inherit WeakReferenceListBase<'T>
        interface IList<'T>
        interface ICollection<'T>
        interface IEnumerable<'T>
        interface IEnumerable
    end
```

Type Parameters

T




The type of objects contained in this kind of list

The WeakReferenceList(T) type exposes the following members.









Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|------------------------------------------------|
|  | WeakReferenceList(T) | Constructs a new list with a presized capacity |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|--------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list (Overrides WeakReferenceListBase(T).Count.) |
|  | Item | Gets or sets an item at a certain index (Overrides WeakReferenceListBase(T).Item(Int32).) |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 4000) (Overrides OptimizedPersistable.ObjectsPerPage.) |


Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add | Adds an item to a list (Overrides WeakReferenceListBase(T).Add(T).) |
|  | AddRange | Adds the elements of the given collection to the end of this list. (Overrides WeakReferenceListBase(T).AddRange(IEnumerable(T)).) |
|  | Clear | Removes all items from the list and frees the array (Overrides WeakReferenceListBase(T).Clear().) |
|  | Insert | Inserts an item at a specified index (Overrides WeakReferenceListBase(T).Insert(Int32, T).) |
|  | InsertRange | Inserts the elements of the given collection at a given index. (Overrides WeakReferenceListBase(T).InsertRange(Int32, IEnumerable(T)).) |
|  | Remove | Remove an item (Overrides WeakReferenceListBase(T).Remove(T).) |
|  | RemoveAt | Removes an item at a specified index (Overrides WeakReferenceListBase(T).RemoveAt(Int32).) |
|  | RemoveRange | Removes a range of items (Overrides WeakReferenceListBase(T).RemoveRange(Int32, Int32).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------|
|  | Get(T) | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|

See Also

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T) Constructor

Constructs a new list with a presized capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakReferenceList(  
    int capacity = 1,  
    SessionBase session = null  
)
```

VB

```
Public Sub New (  
    Optional capacity As Integer = 1,  
    Optional session As SessionBase = Nothing  
)
```

C++

```
public:  
WeakReferenceList(  
    int capacity = 1,  
    SessionBase^ session = nullptr  
)
```

F#

```
new :  
    ?capacity : int *  
    ?session : SessionBase  
(* Defaults:  
    let _capacity = defaultArg capacity 1  
    let _session = defaultArg session null  
)  
-> WeakReferenceList
```

Parameters

capacity (Optional)

Type: [System.Int32](#)

Reserve space for this many elements

session (Optional)

Type: [VelocityDb.Session.SessionBase](#)

Active session

See Also

[WeakReferenceList\(T\)Class](#)




VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).WeakReferenceList(T) Properties

The [WeakReferenceList\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list (Overrides WeakReferenceListBase(T).Count.) |
|  | Item | Gets or sets an item at a certain index (Overrides WeakReferenceListBase(T).Item(Int32.)) |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Count Property

Number of items in list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Count { get; }
```

VB

```
Public Overrides ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () override;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Return Value

Type: [Int32](#)

The number of elements contained in the [ICollection\(T\)](#).

Implements

[ICollection\(T\).Count](#)

[ICollection\(T\).Count](#)

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Item Property

Gets or sets an item at a certain index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override T this[
    int index
] { get; set; }
```

VB

```
Public Overrides Default Property Item (
    index As Integer
) As T
    Get
    Set
```

C++

```
public:
virtual property T default[int index] {
    T get (int index) override;
    void set (int index, T value) override;
}
```

F#

```
abstract Item : 'T with get, set
override Item : 'T with get, set
```

Parameters

index

Type: [System.Int32](#)

the item index

Return Value

Type: *T*

the item at the specified index

Implements

[IList\(T\).Item\(Int32\)](#)

[IList\(T\).Item\(Int32\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|------------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The property is set and the IList(T) is read-only. |

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).ObjectsPerPage Property

Limit to 1000 per page (instead of default 40000)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also









[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)




WeakReferenceList(T).WeakReferenceList(T) Methods

The [WeakReferenceList\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Adds an item to a list (Overrides WeakReferenceListBase(T).Add(T).) |
|  AddRange | Adds the elements of the given collection to the end of this list. (Overrides WeakReferenceListBase(T).AddRange(IEnumerable(T)).) |
|  Clear | Removes all items from the list and frees the array (Overrides WeakReferenceListBase(T).Clear().) |
|  Insert | Inserts an item at a specified index (Overrides WeakReferenceListBase(T).Insert(Int32, T).) |
|  InsertRange | Inserts the elements of the given collection at a given index. (Overrides WeakReferenceListBase(T).InsertRange(Int32, IEnumerable(T)).) |
|  Remove | Remove an item (Overrides WeakReferenceListBase(T).Remove(T).) |
|  RemoveAt | Removes an item at a specified index (Overrides WeakReferenceListBase(T).RemoveAt(Int32).) |
|  RemoveRange | Removes a range of items (Overrides WeakReferenceListBase(T).RemoveRange(Int32, Int32).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Get(T) | (Defined by GraphHelpers.) |
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Add Method

Adds an item to a list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Add(  
    T item  
)
```

VB

```
Public Overrides Sub Add (  
    item As T  
)
```

C++

```
public:  
virtual void Add(  
    T item  
) override
```

F#

```
abstract Add :  
    item : 'T -> unit  
override Add :  
    item : 'T -> unit
```

Parameters

item

Type: *T*

item being added

Implements

[ICollection\(T\).Add\(T\)](#)

[ICollection\(T\).Add\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

See Also

[WeakReferenceList\(T\)Class](#)

VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).AddRange Method

Adds the elements of the given collection to the end of this list.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void AddRange (  
    IEnumerable<T> collection  
)
```

VB

```
Public Overrides Sub AddRange (  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void AddRange (  
    IEnumerable<T>^ collection  
) override
```

F#

```
abstract AddRange :  
    collection : IEnumerable<'T> -> unit  
override AddRange :  
    collection : IEnumerable<'T> -> unit
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be added

See Also

[WeakReferenceList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Clear Method

Removes all items from the list and frees the array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

[ICollection\(T\).Clear\(\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Insert Method

Inserts an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Insert(  
    int index,  
    T item  
)
```

VB

```
Public Overrides Sub Insert (  
    index As Integer,  
    item As T  
)
```

C++

```
public:  
virtual void Insert(  
    int index,  
    T item  
) override
```

F#

```
abstract Insert :  
    index : int *  
    item : 'T -> unit  
override Insert :  
    index : int *  
    item : 'T -> unit
```

Parameters

index

Type: [System.Int32](#)

insert at this index

item

Type: *T*

the item being inserted

Implements

[IList\(T\).Insert\(Int32, T\)](#)

[IList\(T\).Insert\(Int32, T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|---------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The IList(T) is read-only. |

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).InsertRange Method

Inserts the elements of the given collection at a given index.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InsertRange(  
    int index,  
    IEnumerable<T> collection  
)
```

VB

```
Public Overrides Sub InsertRange (  
    index As Integer,  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void InsertRange(  
    int index,  
    IEnumerable<T>^ collection  
) override
```

F#

```
abstract InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit  
override InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit
```

Parameters

index

Type: [System.Int32](#)

start insert at this index

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be inserted

See Also

[WeakReferenceList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).Remove Method

Remove an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Remove (  
    T item  
)
```

VB

```
Public Overrides Function Remove (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    T item  
) override
```

F#

```
abstract Remove :  
    item : 'T -> bool  
override Remove :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to remove

Return Value

Type: [Boolean](#)

true if item was found and removed; otherwise false

Implements

[ICollection\(T\).Remove\(T\)](#)

[ICollection\(T\).Remove\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

VelocityDB Class Library

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).RemoveAt Method

Removes an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RemoveAt (
    int index
)
```

VB

```
Public Overrides Sub RemoveAt (
    index As Integer
)
```

C++

```
public:
virtual void RemoveAt (
    int index
) override
```

F#

```
abstract RemoveAt :
    index : int -> unit
override RemoveAt :
    index : int -> unit
```

Parameters

index

Type: [System.Int32](#)

the index of the item to remove

Implements

[IList\(T\).RemoveAt\(Int32\)](#)

[IList\(T\).RemoveAt\(Int32\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|---------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The IList(T) is read-only. |

VelocityDB Class Library

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceList(T).RemoveRange Method

Removes a range of items

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RemoveRange (  
    int index,  
    int howMany  
)
```

VB

```
Public Overrides Sub RemoveRange (  
    index As Integer,  
    howMany As Integer  
)
```

C++

```
public:  
virtual void RemoveRange (  
    int index,  
    int howMany  
) override
```

F#

```
abstract RemoveRange :  
    index : int *  
    howMany : int -> unit  
override RemoveRange :  
    index : int *  
    howMany : int -> unit
```

Parameters

index

Type: [System.Int32](#)

start index

howMany

Type: [System.Int32](#)

how many to remove

See Also

[WeakReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T) Class

List with an Id containing object Ids ordered by index.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Collection.WeakReferenceListBase(T)

[VelocityDb.Collection.WeakReferenceList\(T\)](#)

[VelocityDb.Collection.WeakShortReferenceList\(T\)](#)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public abstract class WeakReferenceListBase<T> : OptimizedPersistable,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
Public MustInherit Class WeakReferenceListBase(Of T As IOptimizedPersistable)
    Inherits OptimizedPersistable
    Implements IList(Of T), ICollection(Of T),
    IEnumerable(Of T), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename T>
where T : IOptimizedPersistable
public ref class WeakReferenceListBase abstract : public
    OptimizedPersistable,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<AbstractClassAttribute>]
[<SerializableAttribute>]
type WeakReferenceListBase<'T when 'T : IOptimizedPersistable> =
    class
        inherit OptimizedPersistable
        interface IList<'T>
        interface ICollection<'T>
        interface IEnumerable<'T>
        interface IEnumerable
    end
```







Type Parameters

T















The type of objects contained in this kind of list





The WeakReferenceListBase(T) type exposes the following members.

Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-----------------------------------------|
|  | Count | Number of items in list |
|  | IsFixedSize | Always false |
|  | IsReadOnly | Always false |
|  | IsSynchronized | Always false |
|  | Item | Gets or sets an item at a certain index |
|  | SyncRoot | Used for thread safety |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add | Adds an item to a list |
|  | AddRange | Adds the elements of the given collection to the end of this list. |
|  | BinarySearch | Searches the entire sorted list for an element using the default comparer and returns the zero-based index of the element. |
|  | Clear | Removes all items from the list and frees the array |
|  | ClearAndUnpersistContainedObjects | Removes all items from the list and unpersists all contained objects |
|  | Contains | Checks if an item is contained in the list |
|  | CopyTo(Array, Int32) | Copies list to an array |
|  | CopyTo(T[], Int32) | Copies list items to an array |
|  | GetEnumerator | Returns an enumerator that iterates through the collection. |
|  | IndexOf | Finds the first index of an item |
|  | Insert | Inserts an item at a specified index |
|  | InsertRange | Inserts the elements of the given collection at a given index. |
|  | Last | Get the value at the last position in the list |
|  | Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, |

| | | |
|-----------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | SessionBase , Boolean , Boolean , Queue(IOptimizedPersistable).) |
|  | Remove | Remove an item |
|  | RemoveAt | Removes an item at a specified index |
|  | RemoveRange | Removes a range of items |
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | Get(T) | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |







See Also

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).WeakReferenceListBase(T) Properties

The [WeakReferenceListBase\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-----------------------------------------|
|  | Count | Number of items in list |
|  | IsFixedSize | Always false |
|  | IsReadOnly | Always false |
|  | IsSynchronized | Always false |
|  | Item | Gets or sets an item at a certain index |
|  | SyncRoot | Used for thread safety |

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Count Property

Number of items in list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract int Count { get; }
```

VB

```
Public MustOverride ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () abstract;  
}
```

F#

```
abstract Count : int with get
```

Property Value

Type: [Int32](#)

Implements

[ICollection\(T\).Count](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).IsFixedSize Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsFixedSize { get; }
```

VB

```
Public ReadOnly Property IsFixedSize As Boolean  
    Get
```

C++

```
public:  
property bool IsFixedSize {  
    bool get ();  
}
```

F#

```
member IsFixedSize : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).ReadOnly Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ReadOnly { get; }
```

VB

```
Public ReadOnly Property ReadOnly As Boolean  
    Get
```

C++

```
public:  
virtual property bool ReadOnly {  
    bool get () sealed;  
}
```

F#

```
abstract ReadOnly : bool with get  
override ReadOnly : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[ICollection\(T\).ReadOnly](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).IsSynchronized Property

Always false

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsSynchronized { get; }
```

VB

```
Public ReadOnly Property IsSynchronized As Boolean  
    Get
```

C++

```
public:  
property bool IsSynchronized {  
    bool get ();  
}
```

F#

```
member IsSynchronized : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Item Property

Gets or sets an item at a certain index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract T this[
    int index
] { get; set; }
```

VB

```
Public MustOverride Default Property Item (
    index As Integer
) As T
    Get
    Set
```

C++

```
public:
virtual property T default[int index] {
    T get (int index) abstract;
    void set (int index, T value) abstract;
}
```

F#

```
abstract Item : 'T with get, set
```

Parameters

index

Type: [System.Int32](#)

the item index

Return Value

Type: *T*

the item at the specified index

Implements

[IList\(T\).Item\(Int32\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).SyncRoot Property

Used for thread safety

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object SyncRoot { get; }
```

VB

```
Public ReadOnly Property SyncRoot As Object  
    Get
```

C++

```
public:  
property Object^ SyncRoot {  
    Object^ get ();  
}
```

F#

```
member SyncRoot : Object with get
```

Property Value

Type: [Object](#)

See Also



















[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)


WeakReferenceListBase(T).WeakReferenceListBase(T) Methods

The [WeakReferenceListBase\(T\)](#) generic type exposes the following members.



Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Adds an item to a list |
|  AddRange | Adds the elements of the given collection to the end of this list. |
|  BinarySearch | Searches the entire sorted list for an element using the default comparer and returns the zero-based index of the element. |
|  Clear | Removes all items from the list and frees the array |
|  ClearAndUnpersistContainedObjects | Removes all items from the list and unpersists all contained objects |
|  Contains | Checks if an item is contained in the list |
|  CopyTo(Array, Int32) | Copies list to an array |
|  CopyTo(T[], Int32) | Copies list items to an array |
|  GetEnumerator | Returns an enumerator that iterates through the collection. |
|  IndexOf | Finds the first index of an item |
|  Insert | Inserts an item at a specified index |
|  InsertRange | Inserts the elements of the given collection at a given index. |
|  Last | Get the value at the last position in the list |
|  Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)).) |
|  Remove | Remove an item |
|  RemoveAt | Removes an item at a specified index |
|  RemoveRange | Removes a range of items |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|  Get(T) | (Defined by GraphHelpers.) |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Add Method

Adds an item to a list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Add(  
    T item  
)
```

VB

```
Public MustOverride Sub Add (  
    item As T  
)
```

C++

```
public:  
virtual void Add(  
    T item  
) abstract
```

F#

```
abstract Add :  
    item : 'T -> unit
```

Parameters

item

Type: *T*

item being added

Implements

[ICollection\(T\).Add\(T\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).AddRange Method

Adds the elements of the given collection to the end of this list.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void AddRange (  
    IEnumerable<T> collection  
)
```

VB

```
Public MustOverride Sub AddRange (  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void AddRange (  
    IEnumerable<T>^ collection  
) abstract
```

F#

```
abstract AddRange :  
    collection : IEnumerable<'T> -> unit
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be added

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).BinarySearch Method

Searches the entire sorted list for an element using the default comparer and returns the zero-based index of the element.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int BinarySearch(  
    T aKey,  
    out bool isEqual  
)
```

VB

```
Public Function BinarySearch (  
    aKey As T,  
    <OutAttribute> ByRef isEqual As Boolean  
) As Integer
```

C++

```
public:  
int BinarySearch(  
    T aKey,  
    [OutAttribute] bool% isEqual  
)
```

F#

```
member BinarySearch :  
    aKey : 'T *  
    isEqual : bool byref -> int
```

Parameters

aKey

Type: *T*

The object to locate

isEqual

Type: [System.Boolean](#)

True if match is found; otherwise false

Return Value

Type: [Int32](#)

The index of the object if found; otherwise index at which located would be located

VelocityDB Class Library

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Clear Method

Removes all items from the list and frees the array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Clear()
```

VB

```
Public MustOverride Sub Clear
```

C++

```
public:  
virtual void Clear() abstract
```

F#

```
abstract Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).ClearAndUnpersistContainedObjects Method

Removes all items from the list and unpersists all contained objects

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ClearAndUnpersistContainedObjects (
    SessionBase session
)
```

VB

```
Public Overridable Sub ClearAndUnpersistContainedObjects (
    session As SessionBase
)
```

C++

```
public:
virtual void ClearAndUnpersistContainedObjects (
    SessionBase^ session
)
```

F#

```
abstract ClearAndUnpersistContainedObjects :
    session : SessionBase -> unit
override ClearAndUnpersistContainedObjects :
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.Collection.WeakReferenceListBase`1.ClearAndUnpersistContainedObjects(VelocityDb.Session.SessionBase)"]

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Contains Method

Checks if an item is contained in the list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    T item  
)
```

VB

```
Public Function Contains (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    T item  
) sealed
```

F#

```
abstract Contains :  
    item : 'T -> bool  
override Contains :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to look for

Return Value

Type: [Boolean](#)

true if item was found; otherwise false

Implements

[ICollection\(T\).Contains\(T\)](#)



See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).CopyTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------|
|  | CopyTo(Array, Int32) | Copies list to an array |
|  | CopyTo(T[], Int32) | Copies list items to an array |

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).CopyTo Method (Array, Int32)

Copies list to an array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    Array array,  
    int arrayIndex  
)
```

VB

```
Public Sub CopyTo (  
    array As Array,  
    arrayIndex As Integer  
)
```

C++

```
public:  
void CopyTo(  
    Array^ array,  
    int arrayIndex  
)
```

F#

```
member CopyTo :  
    array : Array *  
    arrayIndex : int -> unit
```

Parameters

array

Type: [System.Array](#)

the array to copy to

arrayIndex

Type: [System.Int32](#)

start index in array for copy

See Also

[WeakReferenceListBase\(T\) Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).CopyTo Method (T[], Int32)

Copies list items to an array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    T[] array,  
    int arrayIndex  
)
```

VB

```
Public Sub CopyTo (  
    array As T(),  
    arrayIndex As Integer  
)
```

C++

```
public:  
virtual void CopyTo(  
    array<T>^ array,  
    int arrayIndex  
) sealed
```

F#

```
abstract CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit  
override CopyTo :  
    array : 'T[] *  
    arrayIndex : int -> unit
```

Parameters

array

Type: *T*[]

the array to copy to

arrayIndex

Type: [System.Int32](#)

start array index of copy

Implements

[ICollection\(T\).CopyTo\(T\[\], Int32\)](#)

VelocityDB Class Library

See Also

[WeakReferenceListBase\(T\)Class](#)

[CopyTo Overload](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).GetEnumerator Method

Returns an enumerator that iterates through the collection.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<T> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of T)
```

C++

```
public:  
virtual IEnumerator<T>^ GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<'T>  
override GetEnumerator : unit -> IEnumerator<'T>
```

Return Value

Type: [IEnumerator\(T\)](#)

An enumerator that can be used to iterate through the collection.

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).IndexOf Method

Finds the first index of an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int IndexOf(  
    T item  
)
```

VB

```
Public Function IndexOf (  
    item As T  
) As Integer
```

C++

```
public:  
virtual int IndexOf(  
    T item  
) sealed
```

F#

```
abstract IndexOf :  
    item : 'T -> int  
override IndexOf :  
    item : 'T -> int
```

Parameters

item

Type: *T*

item to look for

Return Value

Type: [Int32](#)

-1 if item was not found or first index of item

Implements

[IList\(T\).IndexOf\(T\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Insert Method

Inserts an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Insert(  
    int index,  
    T item  
)
```

VB

```
Public MustOverride Sub Insert (  
    index As Integer,  
    item As T  
)
```

C++

```
public:  
virtual void Insert(  
    int index,  
    T item  
) abstract
```

F#

```
abstract Insert :  
    index : int *  
    item : 'T -> unit
```

Parameters

index

Type: [System.Int32](#)

insert at this index

item

Type: *T*

the item being inserted

Implements

[IList\(T\).Insert\(Int32, T\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).InsertRange Method

Inserts the elements of the given collection at a given index.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void InsertRange(  
    int index,  
    IEnumerable<T> collection  
)
```

VB

```
Public MustOverride Sub InsertRange (  
    index As Integer,  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void InsertRange(  
    int index,  
    IEnumerable<T>^ collection  
) abstract
```

F#

```
abstract InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit
```

Parameters

index

Type: [System.Int32](#)

start insert at this index

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be inserted

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Last Method

Get the value at the last position in the list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public T Last ()
```

VB

```
Public Function Last As T
```

C++

```
public:  
T Last ()
```

F#

```
member Last : unit -> 'T
```

Return Value

Type: *T*

The value at the last position in the list

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Persist Method

Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _persistRefs = defaultArg persistRefs true
    let _disableFlush = defaultArg disableFlush false
    let _toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
```

```
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of this persistent object

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue<IOptimizedPersistable>\)](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Remove Method

Remove an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract bool Remove (  
    T item  
)
```

VB

```
Public MustOverride Function Remove (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    T item  
) abstract
```

F#

```
abstract Remove :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to remove

Return Value

Type: [Boolean](#)

true if item was found and removed; otherwise false

Implements

[ICollection\(T\).Remove\(T\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).RemoveAt Method

Removes an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public abstract void RemoveAt (  
    int index  
)
```

```
VB  
Public MustOverride Sub RemoveAt (  
    index As Integer  
)
```

```
C++  
public:  
virtual void RemoveAt (  
    int index  
) abstract
```

```
F#  
abstract RemoveAt :  
    index : int -> unit
```

Parameters

index

Type: [System.Int32](#)

the index of the item to remove

Implements

[IList\(T\).RemoveAt\(Int32\)](#)

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).RemoveRange Method

Removes a range of items

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void RemoveRange (  
    int index,  
    int howMany  
)
```

VB

```
Public MustOverride Sub RemoveRange (  
    index As Integer,  
    howMany As Integer  
)
```

C++

```
public:  
virtual void RemoveRange (  
    int index,  
    int howMany  
) abstract
```

F#

```
abstract RemoveRange :  
    index : int *  
    howMany : int -> unit
```

Parameters

index

Type: [System.Int32](#)

start index

howMany

Type: [System.Int32](#)

how many to remove

See Also

[WeakReferenceListBase\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakReferenceListBase(T).Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[WeakReferenceListBase\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T) Class

List with an Id containing object short Ids (page and slot) ordered by index.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.WeakReferenceListBase\(T\)](#)

VelocityDb.Collection.WeakShortReferenceList(T)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class WeakShortReferenceList<T> : WeakReferenceListBase<T>,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
where T : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class WeakShortReferenceList(Of T As IOptimizedPersistable)
    Inherits WeakReferenceListBase(Of T)
    Implements IList(Of T), ICollection(Of T),
    IEnumerable(Of T), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename T>
where T : IOptimizedPersistable
public ref class WeakShortReferenceList : public WeakReferenceListBase<T>,
    IList<T>, ICollection<T>, IEnumerable<T>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type WeakShortReferenceList<'T when 'T : IOptimizedPersistable> =
    class
        inherit WeakReferenceListBase<'T>
        interface IList<'T>
        interface ICollection<'T>
        interface IEnumerable<'T>
        interface IEnumerable
    end
```


Type Parameters

T




The type of objects contained in this kind of list

The WeakShortReferenceList(T) type exposes the following members.









Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------|
|  | WeakShortReferenceList(T) | Constructs a new list with a presized capacity |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list (Overrides WeakReferenceListBase(T).Count.) |
|  | Item | Gets or sets an item at a certain index (Overrides WeakReferenceListBase(T).Item(Int32).) |
|  | ObjectsPerPage | Limit to 10000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |


Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add | Adds an item to a list (Overrides WeakReferenceListBase(T).Add(T).) |
|  | AddRange | Adds the elements of the given collection to the end of this list. (Overrides WeakReferenceListBase(T).AddRange(IEnumerable(T)).) |
|  | Clear | Removes all items from the list and frees the array (Overrides WeakReferenceListBase(T).Clear().) |
|  | Insert | Inserts an item at a specified index (Overrides WeakReferenceListBase(T).Insert(Int32, T).) |
|  | InsertRange | Inserts the elements of the given collection at a given index. (Overrides WeakReferenceListBase(T).InsertRange(Int32, IEnumerable(T)).) |
|  | Remove | Remove an item (Overrides WeakReferenceListBase(T).Remove(T).) |
|  | RemoveAt | Removes an item at a specified index (Overrides WeakReferenceListBase(T).RemoveAt(Int32).) |
|  | RemoveRange | Removes a range of items (Overrides WeakReferenceListBase(T).RemoveRange(Int32, Int32).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------|
|  | Get(T) | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|

See Also

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T) Constructor

Constructs a new list with a presized capacity

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakShortReferenceList(  
    int capacity = 1,  
    SessionBase session = null  
)
```

VB

```
Public Sub New (  
    Optional capacity As Integer = 1,  
    Optional session As SessionBase = Nothing  
)
```

C++

```
public:  
WeakShortReferenceList(  
    int capacity = 1,  
    SessionBase^ session = nullptr  
)
```

F#

```
new :  
    ?capacity : int *  
    ?session : SessionBase  
(* Defaults:  
    let _capacity = defaultArg capacity 1  
    let _session = defaultArg session null  
)  
-> WeakShortReferenceList
```

Parameters

capacity (Optional)

Type: [System.Int32](#)

Reserve space for this many elements

session (Optional)

Type: [VelocityDb.Session.SessionBase](#)

Active session

See Also

[WeakShortReferenceList\(T\)Class](#)




VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).WeakShortReferenceList(T) Properties

The [WeakShortReferenceList\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------|
|  | Count | Number of items in list (Overrides WeakReferenceListBase(T).Count.) |
|  | Item | Gets or sets an item at a certain index (Overrides WeakReferenceListBase(T).Item(Int32.)) |
|  | ObjectsPerPage | Limit to 10000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Count Property

Number of items in list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Count { get; }
```

VB

```
Public Overrides ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () override;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Return Value

Type: [Int32](#)

The number of elements contained in the [ICollection\(T\)](#).

Implements

[ICollection\(T\).Count](#)

[ICollection\(T\).Count](#)

See Also

[WeakShortReferenceList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Item Property

Gets or sets an item at a certain index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override T this[
    int index
] { get; set; }
```

VB

```
Public Overrides Default Property Item (
    index As Integer
) As T
    Get
    Set
```

C++

```
public:
virtual property T default[int index] {
    T get (int index) override;
    void set (int index, T value) override;
}
```

F#

```
abstract Item : 'T with get, set
override Item : 'T with get, set
```

Parameters

index

Type: [System.Int32](#)

the item index

Return Value

Type: *T*

the item at the specified index

Implements

[IList\(T\).Item\(Int32\)](#)

[IList\(T\).Item\(Int32\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|------------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The property is set and the IList(T) is read-only. |

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).ObjectsPerPage Property

Limit to 10000 per page (instead of default 40000)

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also









[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)




WeakShortReferenceList(T).WeakShortReferenceList(T) Methods

The [WeakShortReferenceList\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Adds an item to a list (Overrides WeakReferenceListBase(T).Add(T).) |
|  AddRange | Adds the elements of the given collection to the end of this list. (Overrides WeakReferenceListBase(T).AddRange(IEnumerable(T)).) |
|  Clear | Removes all items from the list and frees the array (Overrides WeakReferenceListBase(T).Clear().) |
|  Insert | Inserts an item at a specified index (Overrides WeakReferenceListBase(T).Insert(Int32, T).) |
|  InsertRange | Inserts the elements of the given collection at a given index. (Overrides WeakReferenceListBase(T).InsertRange(Int32, IEnumerable(T)).) |
|  Remove | Remove an item (Overrides WeakReferenceListBase(T).Remove(T).) |
|  RemoveAt | Removes an item at a specified index (Overrides WeakReferenceListBase(T).RemoveAt(Int32).) |
|  RemoveRange | Removes a range of items (Overrides WeakReferenceListBase(T).RemoveRange(Int32, Int32).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  Get(T) | (Defined by GraphHelpers.) |
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Add Method

Adds an item to a list

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Add(  
    T item  
)
```

VB

```
Public Overrides Sub Add (  
    item As T  
)
```

C++

```
public:  
virtual void Add(  
    T item  
) override
```

F#

```
abstract Add :  
    item : 'T -> unit  
override Add :  
    item : 'T -> unit
```

Parameters

item

Type: *T*

item being added

Implements

[ICollection\(T\).Add\(T\)](#)

[ICollection\(T\).Add\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

See Also

[WeakShortReferenceList\(T\)Class](#)

VelocityDB Class Library

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).AddRange Method

Adds the elements of the given collection to the end of this list.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void AddRange (  
    IEnumerable<T> collection  
)
```

VB

```
Public Overrides Sub AddRange (  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void AddRange (  
    IEnumerable<T>^ collection  
) override
```

F#

```
abstract AddRange :  
    collection : IEnumerable<'T> -> unit  
override AddRange :  
    collection : IEnumerable<'T> -> unit
```

Parameters

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be added

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Clear Method

Removes all items from the list and frees the array

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

[ICollection\(T\).Clear\(\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

See Also

[WeakShortReferenceList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Insert Method

Inserts an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Insert(  
    int index,  
    T item  
)
```

VB

```
Public Overrides Sub Insert (  
    index As Integer,  
    item As T  
)
```

C++

```
public:  
virtual void Insert(  
    int index,  
    T item  
) override
```

F#

```
abstract Insert :  
    index : int *  
    item : 'T -> unit  
override Insert :  
    index : int *  
    item : 'T -> unit
```

Parameters

index

Type: [System.Int32](#)

insert at this index

item

Type: *T*

the item being inserted

Implements

[IList\(T\).Insert\(Int32, T\)](#)

[IList\(T\).Insert\(Int32, T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|---------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The IList(T) is read-only. |

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).InsertRange Method

Inserts the elements of the given collection at a given index.

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InsertRange(  
    int index,  
    IEnumerable<T> collection  
)
```

VB

```
Public Overrides Sub InsertRange (  
    index As Integer,  
    collection As IEnumerable(Of T)  
)
```

C++

```
public:  
virtual void InsertRange(  
    int index,  
    IEnumerable<T>^ collection  
) override
```

F#

```
abstract InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit  
override InsertRange :  
    index : int *  
    collection : IEnumerable<'T> -> unit
```

Parameters

index

Type: [System.Int32](#)

start insert at this index

collection

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

to be inserted

See Also

[WeakShortReferenceList\(T\) Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).Remove Method

Remove an item

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Remove (  
    T item  
)
```

VB

```
Public Overrides Function Remove (  
    item As T  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    T item  
) override
```

F#

```
abstract Remove :  
    item : 'T -> bool  
override Remove :  
    item : 'T -> bool
```

Parameters

item

Type: *T*

the item to remove

Return Value

Type: [Boolean](#)

true if item was found and removed; otherwise false

Implements

[ICollection\(T\).Remove\(T\)](#)

[ICollection\(T\).Remove\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|--------------------------------------------------|
| NotSupportedException | The ICollection(T) is read-only. |

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).RemoveAt Method

Removes an item at a specified index

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RemoveAt (
    int index
)
```

VB

```
Public Overrides Sub RemoveAt (
    index As Integer
)
```

C++

```
public:
virtual void RemoveAt (
    int index
) override
```

F#

```
abstract RemoveAt :
    index : int -> unit
override RemoveAt :
    index : int -> unit
```

Parameters

index

Type: [System.Int32](#)

the index of the item to remove

Implements

[IList\(T\).RemoveAt\(Int32\)](#)

[IList\(T\).RemoveAt\(Int32\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------------|---------------------------------------------------------------------|
| ArgumentOutOfRangeException | <i>index</i> is not a valid index in the IList(T) . |
| NotSupportedException | The IList(T) is read-only. |

VelocityDB Class Library

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)

WeakShortReferenceList(T).RemoveRange Method

Removes a range of items

Namespace: [VelocityDb.Collection](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RemoveRange (  
    int index,  
    int howMany  
)
```

VB

```
Public Overrides Sub RemoveRange (  
    index As Integer,  
    howMany As Integer  
)
```

C++

```
public:  
virtual void RemoveRange (  
    int index,  
    int howMany  
) override
```

F#

```
abstract RemoveRange :  
    index : int *  
    howMany : int -> unit  
override RemoveRange :  
    index : int *  
    howMany : int -> unit
```

Parameters

index

Type: [System.Int32](#)

start index

howMany

Type: [System.Int32](#)

how many to remove

See Also

[WeakShortReferenceList\(T\)Class](#)

[VelocityDb.Collection Namespace](#)



VelocityDb.Collection.BTree Namespace

The `VelocityDb.Collection.BTree` namespace contains classes for the VelocityDb BTrees optimized for persistent storage in VelocityDb databases

Classes

| Class | Description |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AssemblyExtensions | |
| BTreeBase(Key, Value) | Represents a collection of objects that is maintained in sorted order. |
| BTreeByteArray | Wrapper object for a byte[] of data for comparing objects within a BTreeSet. Wrapper is used so that this data can be placed on a separate Page |
| BTreeInternal(Key, Value) | Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values. |
| BTreeInternalBase(Key, Value) | Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values. |
| BTreeLeaf(Key, Value) | Creates a leaf level node in the BTree |
| BTreeLeafBase(Key, Value) | Internal class used with BTreeSet and BTreeMap |
| BTreeMap(Key, Value) | Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values. |
| BTreeMapBase(Key, Value) | Represents a collection of objects that is maintained in sorted order. |
| BTreeMapIterator(Key, Value) | Iterates all the elements of a BTreeSet |
| BTreeMapOidShort(Key, Value) | Represents a collection of objects that is maintained in sorted order. Collection and all objects must be within a single Database (since references uses OidShort persistently) |
| BTreeNode | A BTree consists of a tree of nodes. Each BTree node has this class a base class. |
| BTreeSet(Key) | Represents a collection of objects that is maintained in sorted order. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  BTreeSetIterator(Key) | Iterates all the elements of a BTreeSet |
|  BTreeSetOidShort(Key) | Represents a collection of objects that is maintained in sorted order. Collection and all objects must be within a single Database (since references uses OidShort persistently) |

AssemblyExtensions Class

[Missing <summary> documentation for "T:VelocityDb.Collection.BTree.AssemblyExtensions"]

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Collection.BTree.AssemblyExtensions

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class AssemblyExtensions
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class AssemblyExtensions
```

C++

```
[ExtensionAttribute]  
public ref class AssemblyExtensions abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type AssemblyExtensions = class end
```

The **AssemblyExtensions** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | GetLoadableTypes | |


See Also

[VelocityDb.Collection.BTree Namespace](#)

AssemblyExtensions.AssemblyExtensions Methods

The [AssemblyExtensions](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-------------|
|  | GetLoadableTypes | |

See Also

[AssemblyExtensions Class](#)

[VelocityDb.Collection.BTree Namespace](#)

AssemblyExtensions.GetLoadableTypes Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.AssemblyExtensions.GetLoadableTypes(System.Reflection.Assembly)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEnumerable<Type> GetLoadableTypes (  
    this Assembly assembly  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function GetLoadableTypes (  
    assembly As Assembly  
) As IEnumerable(Of Type)
```

C++

```
public:  
[ExtensionAttribute]  
static IEnumerable<Type^> GetLoadableTypes (  
    Assembly^ assembly  
)
```

F#

```
[<ExtensionAttribute>]  
static member GetLoadableTypes :  
    assembly : Assembly -> IEnumerable<Type>
```

Parameters

assembly

Type: [System.Reflection.Assembly](#)

[Missing <param name="assembly"/> documentation for

"M:VelocityDb.Collection.BTree.AssemblyExtensions.GetLoadableTypes(System.Reflection.Assembly)"]

Return Value

Type: [IEnumerable\(Type\)](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.BTree.AssemblyExtensions.GetLoadableTypes(System.Reflection.Assembly)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Assembly](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[AssemblyExtensions Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value) Class

Represents a collection of objects that is maintained in sorted order.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

VelocityDb.Collection.BTree.BTreeBase(Key, Value)

[VelocityDb.Collection.BTree.BTreeMapBase\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeSet\(Key\)](#)

[VelocityDb.Collection.BTree.BTreeSetOidShort\(Key\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class BTreeBase<Key, Value> : BTreeNode,  
    IEnumerable<Key>, IEnumerable
```

VB

```
<SerializableAttribute>  
Public MustInherit Class BTreeBase(Of Key, Value)  
    Inherits BTreeNode  
    Implements IEnumerable(Of Key), IEnumerable
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class BTreeBase abstract : public BTreeNode,  
    IEnumerable<Key>, IEnumerable
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type BTreeBase<'Key, 'Value> =  
    class  
        inherit BTreeNode  
        interface IEnumerable<'Key>  
        interface IEnumerable  
    end
```

Type Parameters

Key














The type of key objects in this BTree

Value





The type of value objects in a BTreeMap or BTreeMapOidShort



















The BTreeBase(Key, Value) type exposes the following members.

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer(Of Key) that is used to determine the order of the items in the sorted set |
|  | ComparisonArraySize | Gets the size of a comparison array used by this set. |
|  | Count | Gets the number of elements in the set. |
|  | Depth | Gets the depth of the BTreeSet |
|  | FlushIfPageFull | We don't want to flush other BTree pages while looking for a placement page (Overrides OptimizedPersistable.FlushIfPageFull.) |
|  | GetAlternateComparer | If comparer isn't set then if Key type is string use SessionBase.DefaultStringComparer or ... |
|  | IsLeaf | Is the root a leaf node or not? |
|  | KeyPlacement | Set the key placement to be used for all key objects added to this. This setting is not persisted, it is mainly to be used with objects added with AddFast(Key) |
|  | NodeSize | The size of the nodeList or the keyList (if depth is 1) of the root node. |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | ToDoBatchAddCount | Gets the number of objects that are not yet added to the collection |
|  | TransientBatchSize | Get/Set the max batch size used in API such as AddFast(Key) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add(Key) | Adds an element to the set and returns a value that indicates if it was successfully added |
|  | Add(Key, Value) | Add a key with an associated value to a BTreeMap |
|  | Add(Key, Value, Byte[]) | Adds a key and value to the map and returns a value that indicates if it was successfully added |
|  | AddFast | Adds an element to an array of to be added objects. The objects in this array are added when a call to FlushTransients() is triggered by multiple events such as calling Count and when collection is committed and/or flushed to disk or when the array is full. The array size is by default |

| | | |
|-------------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | currently BTreeAddFastTransientBatchSize but you can get/set the desired size with TransientBatchSize |
|  | AddWithComparisonArray | Adds an element to the set and returns a value that indicates if it was successfully added |
|  | Clear | Removes all elements from the set. |
|  | Contains(Key) | Determines whether the set contains a specific element. |
|  | Contains(Key,Byte[]) | Determines whether the set contains a specific element. |
|  | CopyTo | Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array. |
|  | FlushTransients | Adds all queued up to be added objects after presorting them transiently (Overrides OptimizedPersistable.FlushTransients().) |
|  | GetEnumerator | Enumerates all contained Key objects in sorted order |
|  | GetKeyId(Key) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |
|  | GetKeyId(Key,Byte[]) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |
|  | InitializeAfterRead | Sets up some transient variables (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  | InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  | InitNew | Internal use only |
|  | Iterator | Initializes an iterator to find the keys of this set |
|  | Remove(Key) | Removes a specified item from the set. |
|  | Remove(Key,Byte[]) | Removes a specified item from the set. |
|  | SearchTransients | |
|  | TryGetKey(Key, Key) | Gets the value matching persistent key with the specified key. |
|  | TryGetKey(Key,Byte[], Key) | Gets the value associated with the specified key. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |














See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).BTreeBase(Key, Value) Properties

The [BTreeBase\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer(Of Key) that is used to determine the order of the items in the sorted set |
|  | ComparisonArraySize | Gets the size of a comparison array used by this set. |
|  | Count | Gets the number of elements in the set. |
|  | Depth | Gets the depth of the BTreeSet |
|  | FlushIfPageFull | We don't want to flush other BTree pages while looking for a placement page (Overrides OptimizedPersistable.FlushIfPageFull.) |
|  | GetAlternateComparer | If comparer isn't set then if Key type is string use SessionBase.DefaultStringComparer or ... |
|  | IsLeaf | Is the root a leaf node or not? |
|  | KeyPlacement | Set the key placement to be used for all key objects added to this. This setting is not persisted, it is mainly to be used with objects added with AddFast(Key) |
|  | NodeSize | The size of the nodeList or the keyList (if depth is 1) of the root node. |
|  | ObjectsPerPage | Limit to 1000 per page (instead of default 40000) (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | ToDoBatchAddCount | Gets the number of objects that are not yet added to the collection |
|  | TransientBatchSize | Get/Set the max batch size used in API such as AddFast(Key) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Comparer Property

Gets the VelocityDbComparer(Of Key) that is used to determine the order of the items in the sorted set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public MustOverride ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () abstract;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).ComparisonArraySize Property

Gets the size of a comparison array used by this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort ComparisonArraySize { get; }
```

VB

```
Public ReadOnly Property ComparisonArraySize As UShort  
    Get
```

C++

```
public:  
property unsigned short ComparisonArraySize {  
    unsigned short get ();  
}
```

F#

```
member ComparisonArraySize : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Count Property

Gets the number of elements in the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get () sealed;  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Depth Property

Gets the depth of the BTreeSet

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort Depth { get; }
```

VB

```
Public ReadOnly Property Depth As UShort  
    Get
```

C++

```
public:  
property unsigned short Depth {  
    unsigned short get ();  
}
```

F#

```
member Depth : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).FlushIfPageFull Property

We don't want to flush other BTree pages while looking for a placement page

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool FlushIfPageFull { get; }
```

VB

```
Public Overrides ReadOnly Property FlushIfPageFull As Boolean  
    Get
```

C++

```
public:  
virtual property bool FlushIfPageFull {  
    bool get () override;  
}
```

F#

```
abstract FlushIfPageFull : bool with get  
override FlushIfPageFull : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.FlushIfPageFull](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).GetAlternateComparer Property

If comparer isn't set then if Key type is string use SessionBase.DefaultStringComparer or ...

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IComparer<Key> GetAlternateComparer { get; }
```

VB

```
Public ReadOnly Property GetAlternateComparer As IComparer(Of Key)  
    Get
```

C++

```
public:  
property IComparer<Key>^ GetAlternateComparer {  
    IComparer<Key>^ get ();  
}
```

F#

```
member GetAlternateComparer : IComparer<'Key> with get
```

Property Value

Type: [IComparer](#)(Key)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).IsLeaf Property

Is the root a leaf node or not?

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsLeaf { get; }
```

VB

```
Public ReadOnly Property IsLeaf As Boolean  
    Get
```

C++

```
public:  
property bool IsLeaf {  
    bool get ();  
}
```

F#

```
member IsLeaf : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).KeyPlacement Property

Set the key placement to be used for all key objects added to this. This setting is not persisted, it is mainly to be used with objects added with [AddFast\(Key\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Placement KeyPlacement { get; set; }
```

VB

```
Public Property KeyPlacement As Placement  
    Get  
    Set
```

C++

```
public:  
property Placement^ KeyPlacement {  
    Placement^ get ();  
    void set (Placement^ value);  
}
```

F#

```
member KeyPlacement : Placement with get, set
```

Property Value

Type: [Placement](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).NodeSize Property

The size of the nodeList or the keyList (if depth is 1) of the root node.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int NodeSize { get; }
```

VB

```
Public ReadOnly Property NodeSize As Integer  
    Get
```

C++

```
public:  
property int NodeSize {  
    int get ();  
}
```

F#

```
member NodeSize : int with get
```

Property Value

Type: [Int32](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).ObjectsPerPage Property

Limit to 1000 per page (instead of default 40000)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).ToDoBatchAddCount Property

Gets the number of objects that are not yet added to the collection

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ToDoBatchAddCount { get; }
```

VB

```
Public ReadOnly Property ToDoBatchAddCount As UInteger  
    Get
```

C++

```
public:  
property unsigned int ToDoBatchAddCount {  
    unsigned int get ();  
}
```

F#

```
member ToDoBatchAddCount : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).TransientBatchSize Property

Get/Set the max batch size used in API such as [AddFast\(Key\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual uint TransientBatchSize { get; set; }
```

VB

```
Public Overridable Property TransientBatchSize As UInteger  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned int TransientBatchSize {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
abstract TransientBatchSize : uint32 with get, set  
override TransientBatchSize : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).UsesOidShort Property

Does this BTree use short object references (32 bit) for its internal references?

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract bool UsesOidShort { get; }
```

VB

```
Public MustOverride ReadOnly Property UsesOidShort As Boolean  
    Get
```

C++

```
public:  
virtual property bool UsesOidShort {  
    bool get () abstract;  
}
```

F#

```
abstract UsesOidShort : bool with get
```

Property Value

Type: [Boolean](#)

See Also
















[BTreeBase\(Key, Value\)Class](#)








[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).BTreeBase(Key, Value) Methods



The [BTreeBase\(Key, Value\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add(Key) | Adds an element to the set and returns a value that indicates if it was successfully added |
|  Add(Key, Value) | Add a key with an associated value to a BTreeMap |
|  Add(Key, Value,Byte[]) | Adds a key and value to the map and returns a value that indicates if it was successfully added |
|  AddFast | Adds an element to an array of to be added objects. The objects in this array are added when a call to FlushTransients() is triggered by multiple events such as calling Count and when collection is committed and/or flushed to disk or when the array is full. The array size is by default currently BTreeAddFastTransientBatchSize but you can get/set the desired size with TransientBatchSize |
|  AddWithComparisonArray | Adds an element to the set and returns a value that indicates if it was successfully added |
|  Clear | Removes all elements from the set. |
|  Contains(Key) | Determines whether the set contains a specific element. |
|  Contains(Key,Byte[]) | Determines whether the set contains a specific element. |
|  CopyTo | Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array. |
|  FlushTransients | Adds all queued up to be added objects after presorting them transiently (Overrides OptimizedPersistable.FlushTransients().) |
|  GetEnumerator | Enumerates all contained Key objects in sorted order |
|  GetKeyId(Key) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |
|  GetKeyId(Key,Byte[]) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |
|  InitializeAfterRead | Sets up some transient variables (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide |

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| | | your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  | InitNew | Internal use only |
|  | Iterator | Initializes an iterator to find the keys of this set |
|  | Remove(Key) | Removes a specified item from the set. |
|  | Remove(Key,Byte[]) | Removes a specified item from the set. |
|  | SearchTransients | |
|  | TryGetKey(Key, Key) | Gets the value matching persistent key with the specified key. |
|  | TryGetKey(Key,Byte[], Key) | Gets the value associated with the specified key. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |




See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Add Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------|
|  | Add(Key) | Adds an element to the set and returns a value that indicates if it was successfully added |
|  | Add(Key, Value) | Add a key with an associated value to a BTreeMap |
|  | Add(Key, Value, Byte[]) | Adds a key and value to the map and returns a value that indicates if it was successfully added |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Add Method (Key)

Adds an element to the set and returns a value that indicates if it was successfully added

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Add(  
    Key key  
)
```

VB

```
Public Function Add (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Add(  
    Key key  
) sealed
```

F#

```
abstract Add :  
    key : 'Key -> bool  
override Add :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

The object being added

Return Value

Type: [Boolean](#)

true if an element was added; otherwise, false.

See Also

[BTreeBase\(Key, Value\)Class](#)

[Add Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Add Method (Key, Value)

Add a key with an associated value to a BTreeMap

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Add(  
    Key key,  
    Value value  
)
```

VB

```
Public Sub Add (  
    key As Key,  
    value As Value  
)
```

C++

```
public:  
virtual void Add(  
    Key key,  
    Value value  
) sealed
```

F#

```
abstract Add :  
    key : 'Key *  
    value : 'Value -> unit  
override Add :  
    key : 'Key *  
    value : 'Value -> unit
```

Parameters

key

Type: *Key*

The key added

value

Type: *Value*

The associated value added

See Also

[BTreeBase\(Key, Value\)Class](#)

[Add Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Add Method (Key, Value, Byte[])

Adds a key and value to the map and returns a value that indicates if it was successfully added

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Add(  
    Key key,  
    Value value,  
    byte[] comparisonArrayIn  
)
```

VB

```
Public Sub Add (  
    key As Key,  
    value As Value,  
    comparisonArrayIn As Byte()  
)
```

C++

```
public:  
void Add(  
    Key key,  
    Value value,  
    array<unsigned char>^ comparisonArrayIn  
)
```

F#

```
member Add :  
    key : 'Key *  
    value : 'Value *  
    comparisonArrayIn : byte[] -> unit
```

Parameters

key

Type: *Key*

[Missing <param name="key"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Add(`0,`1,System.Byte[])"]

value

Type: *Value*

[Missing <param name="value"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Add(`0,`1,System.Byte[])"]

comparisonArrayIn

Type: [System.Byte\[\]](#)

[Missing <param name="comparisonArrayIn"/> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.Add(`0,`1,System.Byte[])"]

Return Value

Type:

true if an element was added; otherwise, false.

See Also

[BTreeBase\(Key, Value\)Class](#)

[Add Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).AddFast Method

Adds an element to an array of to be added objects. The objects in this array are added when a call to [FlushTransients\(\)](#) is triggered by multiple events such as calling [Count](#) and when collection is committed and/or flushed to disk or when the array is full. The array size is by default currently [BTreeAddFastTransientBatchSize](#) but you can get/set the desired size with [TransientBatchSize](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddFast (  
    Key key  
)
```

VB

```
Public Sub AddFast (  
    key As Key  
)
```

C++

```
public:  
void AddFast (  
    Key key  
)
```

F#

```
member AddFast :  
    key : 'Key -> unit
```

Parameters

key

Type: *Key*

The object being added

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).AddWithComparisonArray Method

Adds an element to the set and returns a value that indicates if it was successfully added

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AddWithComparisonArray(  
    Key key,  
    byte[] comparisonArrayIn  
)
```

VB

```
Public Function AddWithComparisonArray (  
    key As Key,  
    comparisonArrayIn As Byte ()  
) As Boolean
```

C++

```
public:  
bool AddWithComparisonArray(  
    Key key,  
    array<unsigned char>^ comparisonArrayIn  
)
```

F#

```
member AddWithComparisonArray :  
    key : 'Key *  
    comparisonArrayIn : byte[] -> bool
```

Parameters

key

Type: *Key*

[Missing <param name="key"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.AddWithComparisonArray(`0,System.Byte[])"]

comparisonArrayIn

Type: [System.Byte\[\]](#)

[Missing <param name="comparisonArrayIn"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.AddWithComparisonArray(`0,System.Byte[])"]

Return Value

Type: [Boolean](#)

true if an element was added; otherwise, false.

VelocityDB Class Library

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Clear Method

Removes all elements from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Clear ()
```

VB

```
Public Overridable Sub Clear
```

C++

```
public:  
virtual void Clear ()
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```



See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Contains Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------|
|  | Contains(Key) | Determines whether the set contains a specific element. |
|  | Contains(Key,Byte[]) | Determines whether the set contains a specific element. |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Contains Method (Key)

Determines whether the set contains a specific element.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Key key  
)
```

VB

```
Public Function Contains (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Key key  
) sealed
```

F#

```
abstract Contains :  
    key : 'Key -> bool  
override Contains :  
    key : 'Key -> bool
```

Parameters

key

Type: Key

[Missing <param name="key"/> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.Contains(`0)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.Contains(`0)"]

See Also

[BTreeBase\(Key, Value\)Class](#)

[Contains Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Contains Method (Key, Byte[])

Determines whether the set contains a specific element.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Key key,  
    byte[] comparisonArray  
)
```

VB

```
Public Function Contains (  
    key As Key,  
    comparisonArray As Byte ()  
) As Boolean
```

C++

```
public:  
bool Contains(  
    Key key,  
    array<unsigned char>^ comparisonArray  
)
```

F#

```
member Contains :  
    key : 'Key *  
    comparisonArray : byte[] -> bool
```

Parameters

key

Type: *Key*

[Missing <param name="key"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Contains(`0,System.Byte[])"]

comparisonArray

Type: [System.Byte\[\]](#)

[Missing <param name="comparisonArray"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Contains(`0,System.Byte[])"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Contains(`0,System.Byte[])"]

VelocityDB Class Library

See Also

[BTreeBase\(Key, Value\)Class](#)

[Contains Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).CopyTo Method

Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    Key[] array,  
    int arrayIndex,  
    int count  
)
```

VB

```
Public Sub CopyTo (  
    array As Key(),  
    arrayIndex As Integer,  
    count As Integer  
)
```

C++

```
public:  
void CopyTo(  
    array<Key>^ array,  
    int arrayIndex,  
    int count  
)
```

F#

```
member CopyTo :  
    array : 'Key[] *  
    arrayIndex : int *  
    count : int -> unit
```

Parameters

array

Type: *Key[]*

The one-dimensional Array that is the destination of the elements copied from BTreeBase. The Array must have zero-based indexing.

arrayIndex

Type: [System.Int32](#)

The zero-based index in the array at which copying begins.

count

VelocityDB Class Library

Type: [System.Int32](#)

The number of elements to copy.

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).FlushTransients Method

Adds all queued up to be added objects after presorting them transiently

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushTransients ()
```

VB

```
Public Overrides Sub FlushTransients
```

C++

```
public:  
virtual void FlushTransients () override
```

F#

```
abstract FlushTransients : unit -> unit  
override FlushTransients : unit -> unit
```

Implements

[IOptimizedPersistable.FlushTransients\(\)](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).GetEnumerator Method

Enumerates all contained Key objects in sorted order

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Key> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of Key)
```

C++

```
public:  
virtual IEnumerable<Key>^ GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<'Key>  
override GetEnumerator : unit -> IEnumerable<'Key>
```

Return Value

Type: [IEnumerable\(Key\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)



See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).GetKeyId Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetKeyId(Key) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |
|  | GetKeyId(Key,Byte[]) | When Key type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used. |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).GetKeyId Method (Key)

When Key type implements [IOptimizedPersistable](#), you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong GetKeyId(  
    Key key  
)
```

VB

```
Public Function GetKeyId (  
    key As Key  
) As ULong
```

C++

```
public:  
unsigned long long GetKeyId(  
    Key key  
)
```

F#

```
member GetKeyId :  
    key : 'Key -> uint64
```

Parameters

key

Type: *Key*

Transient lookup object of type *Key* (which in this case must be implementing [IOptimizedPersistable](#))

Return Value

Type: [UInt64](#)

The Id of the corresponding persistent object or 0 if not persistent or not implementing [IOptimizedPersistable](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[GetKeyId Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).GetKeyId Method (Key, Byte[])

When Key type implements [IOptimizedPersistable](#), you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. To avoid opening the object, a comparison array which is flagged as complete key must be used.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong GetKeyId(
    Key key,
    byte[] comparisonArray
)
```

VB

```
Public Function GetKeyId (
    key As Key,
    comparisonArray As Byte()
) As ULong
```

C++

```
public:
    unsigned long long GetKeyId(
        Key key,
        array<unsigned char>^ comparisonArray
    )
```

F#

```
member GetKeyId :
    key : 'Key *
    comparisonArray : byte[] -> uint64
```

Parameters

key

Type: *Key*

Transient lookup object of type *Key* (which in this case must be implementing [IOptimizedPersistable](#))

comparisonArray

Type: [System.Byte\[\]](#)

A byte containing bytes to be compared with.

VelocityDB Class Library

Return Value

Type: [UInt64](#)

The Id of the corresponding persistent object or 0 if not persistent or not implementing [IOptimizedPersistable](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[GetKeyId Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).InitializeAfterRead Method

Sets up some transient variables

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).InitNew Method

Internal use only

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InitNew(  
    SessionBase session,  
    ushort maxEntriesPerNode,  
    ushort comparisonArraySize,  
    bool comparisonArrayIsCompleteKey,  
    bool isMap  
)
```

VB

```
Public Sub InitNew (  
    session As SessionBase,  
    maxEntriesPerNode As UShort,  
    comparisonArraySize As UShort,  
    comparisonArrayIsCompleteKey As Boolean,  
    isMap As Boolean  
)
```

C++

```
public:  
void InitNew(  
    SessionBase^ session,  
    unsigned short maxEntriesPerNode,  
    unsigned short comparisonArraySize,  
    bool comparisonArrayIsCompleteKey,  
    bool isMap  
)
```

F#

```
member InitNew :  
    session : SessionBase *  
    maxEntriesPerNode : uint16 *  
    comparisonArraySize : uint16 *  
    comparisonArrayIsCompleteKey : bool *  
    isMap : bool -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.InitNew(VelocityDb.Session.SessionBase,System.UInt16,
System.UInt16,System.Boolean,System.Boolean)"]

maxEntriesPerNode

Type: [System.UInt16](#)

[Missing <param name="maxEntriesPerNode"/> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.InitNew(VelocityDb.Session.SessionBase,System.UInt16,
System.UInt16,System.Boolean,System.Boolean)"]

comparisonArraySize

Type: [System.UInt16](#)

[Missing <param name="comparisonArraySize"/> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.InitNew(VelocityDb.Session.SessionBase,System.UInt16,
System.UInt16,System.Boolean,System.Boolean)"]

comparisonArrayIsCompleteKey

Type: [System.Boolean](#)

[Missing <param name="comparisonArrayIsCompleteKey"/> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.InitNew(VelocityDb.Session.SessionBase,System.UInt16,
System.UInt16,System.Boolean,System.Boolean)"]

isMap

Type: [System.Boolean](#)

[Missing <param name="isMap"/> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.InitNew(VelocityDb.Session.SessionBase,System.UInt16,
System.UInt16,System.Boolean,System.Boolean)"]

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Iterator Method

Initializes an iterator to find the keys of this set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual BTreeSetIterator<Key> Iterator ()
```

VB

```
Public Overridable Function Iterator As BTreeSetIterator (Of Key)
```

C++

```
public:  
virtual BTreeSetIterator<Key>^ Iterator ()
```

F#

```
abstract Iterator : unit -> BTreeSetIterator<'Key>  
override Iterator : unit -> BTreeSetIterator<'Key>
```

Return Value

Type: [BTreeSetIterator\(Key\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.Iterator"]



See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Remove Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|----------------------------------------|
|  | Remove(Key) | Removes a specified item from the set. |
|  | Remove(Key,Byte[]) | Removes a specified item from the set. |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Remove Method (Key)

Removes a specified item from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Remove (  
    Key key  
)
```

VB

```
Public Function Remove (  
    key As Key  
) As Boolean
```

C++

```
public:  
virtual bool Remove (  
    Key key  
) sealed
```

F#

```
abstract Remove :  
    key : 'Key -> bool  
override Remove :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

[Missing <param name="key"/> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.Remove(`0)"]

Return Value

Type: [Boolean](#)

true if an element was removed; otherwise, false.

See Also

[BTreeBase\(Key, Value\)Class](#)

[Remove Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).Remove Method (Key, Byte[])

Removes a specified item from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Remove(  
    Key key,  
    byte[] comparisonArray  
)
```

VB

```
Public Function Remove (  
    key As Key,  
    comparisonArray As Byte ()  
) As Boolean
```

C++

```
public:  
bool Remove (  
    Key key,  
    array<unsigned char>^ comparisonArray  
)
```

F#

```
member Remove :  
    key : 'Key *  
    comparisonArray : byte[] -> bool
```

Parameters

key

Type: *Key*

[Missing <param name="key"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Remove(`0,System.Byte[])"]

comparisonArray

Type: [System.Byte\[\]](#)

[Missing <param name="comparisonArray"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeBase`2.Remove(`0,System.Byte[])"]

Return Value

Type: [Boolean](#)

true if an element was removed; otherwise, false.

VelocityDB Class Library

See Also

[BTreeBase\(Key, Value\)Class](#)

[Remove Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).SearchTransients Method

[Missing <summary> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.SearchTransients(`0,`0@)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool SearchTransients(  
    Key key,  
    ref Key value  
)
```

VB

```
Public Overridable Function SearchTransients (  
    key As Key,  
    ByRef value As Key  
) As Boolean
```

C++

```
public:  
virtual bool SearchTransients(  
    Key key,  
    Key% value  
)
```

F#

```
abstract SearchTransients :  
    key : 'Key *  
    value : 'Key byref -> bool  
override SearchTransients :  
    key : 'Key *  
    value : 'Key byref -> bool
```

Parameters

key

Type: Key

[Missing <param name="key"/> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.SearchTransients(`0,`0@)"]

value

Type: Key

[Missing <param name="value"/> documentation for "M:VelocityDb.Collection.BTree.BTreeBase`2.SearchTransients(`0,`0@)"]

VelocityDB Class Library

Return Value

Type: [Boolean](#)

**[Missing <returns> documentation for
"M:VelocityDb.Collection.BTree.BTreeBase`2.SearchTransients(`0,`0@)"]**



See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).TryGetKey Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------------------------|
|  | TryGetKey(Key, Key) | Gets the value matching persistent key with the specified key. |
|  | TryGetKey(Key,Byte[], Key) | Gets the value associated with the specified key. |

See Also

[BTreeBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).TryGetKey Method (Key, Key)

Gets the value matching persistent key with the specified key.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetKey(  
    Key key,  
    ref Key value  
)
```

VB

```
Public Function TryGetKey (  
    key As Key,  
    ByRef value As Key  
) As Boolean
```

C++

```
public:  
bool TryGetKey(  
    Key key,  
    Key% value  
)
```

F#

```
member TryGetKey :  
    key : 'Key *  
    value : 'Key byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

value

Type: *Key*

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the set contains an element with the specified key; otherwise, false.

VelocityDB Class Library

See Also

[BTreeBase\(Key, Value\)Class](#)

[TryGetKey Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeBase(Key, Value).TryGetKey Method (Key, Byte[], Key)

Gets the value associated with the specified key.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetKey(  
    Key key,  
    byte[] comparisonArray,  
    ref Key value  
)
```

VB

```
Public Function TryGetKey (  
    key As Key,  
    comparisonArray As Byte(),  
    ByRef value As Key  
) As Boolean
```

C++

```
public:  
bool TryGetKey(  
    Key key,  
    array<unsigned char>^ comparisonArray,  
    Key% value  
)
```

F#

```
member TryGetKey :  
    key : 'Key *  
    comparisonArray : byte[] *  
    value : 'Key byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

comparisonArray

Type: [System.Byte\[\]](#)

The comparison array associated with the key to get.

value

Type: *Key*

VelocityDB Class Library

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the BTreeBase(Of Key) contains an element with the specified key; otherwise, false.

See Also

[BTreeBase\(Key, Value\)Class](#)

[TryGetKey Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray Class

Wrapper object for a byte[] of data for comparing objects within a BTreeSet. Wrapper is used so that this data can be placed on a separate [Page](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Collection.BTree.BTreeByteArray

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
[SerializableAttribute]
public class BTreeByteArray : OptimizedPersistable
```



```
VB
<SerializableAttribute>
Public Class BTreeByteArray
    Inherits OptimizedPersistable
```



```
C++
[SerializableAttribute]
public ref class BTreeByteArray : public OptimizedPersistable
```

```
F#
[<SerializableAttribute>]
type BTreeByteArray =
    class
        inherit OptimizedPersistable
    end
```


The **BTreeByteArray** type exposes the following members.

Properties



| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache.) |

| | |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
|  ObjectsPerPage | Place this type of object on its own page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  PagesPerDatabase | Use the max limit for this type (Overrides OptimizedPersistable.PagesPerDatabase.) |

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |





See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.BTreeByteArray Properties

The [BTreeByteArray](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache.) |
|  | ObjectsPerPage | Place this type of of object on its own page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | PagesPerDatabase | Use the max limit for this type (Overrides OptimizedPersistable.PagesPerDatabase.) |

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.AllowOtherTypesOnSamePage Property

We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.Cache Property

We do want to cache this kind of object since they are likely to be frequently used.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.ObjectsPerPage Property

Place this type of object on its own page

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override ushort ObjectsPerPage { get; }
```

```
VB  
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

```
C++  
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

```
F#  
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Return Value

Type: [UInt16](#)

The default maximum number of objects per page

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.PagesPerDatabase Property

Use the max limit for this type

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort PagesPerDatabase { get; }
```

VB

```
Public Overrides ReadOnly Property PagesPerDatabase As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short PagesPerDatabase {  
    unsigned short get () override;  
}
```

F#

```
abstract PagesPerDatabase : uint16 with get  
override PagesPerDatabase : uint16 with get
```

Return Value

Type: [UInt16](#)

The chosen limit

Implements

[IOptimizedPersistable.PagesPerDatabase](#)

See Also


[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeByteArray.BTreeByteArray Methods

The [BTreeByteArray](#) type exposes the following members.

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[],Int32,SessionBase,Page,Boolean,Schema,Boolean,List(IOptimizedPersistable),Int32,Int32,Boolean).) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeByteArray.ReadMe Method

Used by code generator

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
```

```
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

memberBytes

Type: [System.Byte\[\]](#)

offset

Type: [System.Int32](#)

session

Type: [VelocityDb.Session.SessionBase](#)

page

Type: [VelocityDb.Page](#)

useOidShort

Type: [System.Boolean](#)

schema

Type: [VelocityDb.TypeInfo.Schema](#)

VelocityDB Class Library

openRefs

Type: [System.Boolean](#)

toLoadMembers

Type: [System.Collections.Generic.List\(IOptimizedPersistable\)](#)

graphDepth

Type: [System.Int32](#)

graphDepthToLoad

Type: [System.Int32](#)

primitivesOnly

Type: [System.Boolean](#)

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List\(IOptimizedPersistable\), Int32, Int32, Boolean\)](#)

See Also

[BTreeByteArray Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternal(Key, Value) Class

Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeInternalBase\(Key, Value\)](#)

VelocityDb.Collection.BTree.BTreeInternal(Key, Value)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class BTreeInternal<Key, Value> : BTreeInternalBase<Key, Value>
```

VB

```
<SerializableAttribute>
Public Class BTreeInternal(Of Key, Value)
    Inherits BTreeInternalBase(Of Key, Value)
```

C++

```
[SerializableAttribute]
generic<typename Key, typename Value>
public ref class BTreeInternal : public BTreeInternalBase<Key, Value>
```

F#

```
[<SerializableAttribute>]
type BTreeInternal<'Key, 'Value> =
    class
        inherit BTreeInternalBase<'Key, 'Value>
    end
```

Type Parameters

Key

The key type of this collection

Value



The value type of this collection

The BTreeInternal(Key, Value) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------------------------------------------|
|  | InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.BTree Namespace](#)



BTreeInternal(Key, Value).BTreeInternal(Key, Value) Methods

The [BTreeInternal\(Key, Value\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------------------------------------------|
|  | InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[BTreeInternal\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternal(Key, Value).InitializeAfterRead Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.BTreeInternal`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"
]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeInternal`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"
]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeInternal\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternalBase(Key, Value) Class

Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

VelocityDb.Collection.BTree.BTreeInternalBase(Key, Value)

[VelocityDb.Collection.BTree.BTreeInternal\(Key, Value\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class BTreeInternalBase<Key, Value> : BTreeNode
```

VB

```
<SerializableAttribute>  
Public MustInherit Class BTreeInternalBase(Of Key, Value)  
    Inherits BTreeNode
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class BTreeInternalBase abstract : public BTreeNode
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type BTreeInternalBase<'Key, 'Value> =  
    class  
        inherit BTreeNode  
    end
```

Type Parameters

Key



The key type of this collection

Value



The value type of this collection

The BTreeInternalBase(Key, Value) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache .) |
|  | NodeSize | Gets size of node (how many child nodes this node has) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |



See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternalBase(Key, Value).BTreeInternalBase(Key, Value) Properties

The [BTreeInternalBase\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | We do want to cache this kind of object since they are likely to be frequently used. (Overrides OptimizedPersistable.Cache.) |
|  | NodeSize | Gets size of node (how many child nodes this node has) |

See Also

[BTreeInternalBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternalBase(Key, Value).Cache Property

We do want to cache this kind of object since they are likely to be frequently used.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[BTreeInternalBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternalBase(Key, Value).NodeSize Property

Gets size of node (how many child nodes this node has)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int NodeSize { get; }
```

VB

```
Public ReadOnly Property NodeSize As Integer  
    Get
```

C++

```
public:  
property int NodeSize {  
    int get ();  
}
```

F#

```
member NodeSize : int with get
```

Property Value

Type: [Int32](#)

See Also



[BTreeInternalBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeInternalBase(Key, Value).BTreeInternalBase(Key, Value) Methods

The [BTreeInternalBase\(Key, Value\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[BTreeInternalBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeaf(Key, Value) Class

Creates a leaf level node in the BTree

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeLeafBase\(Key, Value\)](#)

VelocityDb.Collection.BTree.BTreeLeaf(Key, Value)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class BTreeLeaf<Key, Value> : BTreeLeafBase<Key, Value>
```

VB

```
<SerializableAttribute>  
Public Class BTreeLeaf(Of Key, Value)  
    Inherits BTreeLeafBase(Of Key, Value)
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class BTreeLeaf : public BTreeLeafBase<Key, Value>
```

F#

```
[<SerializableAttribute>]  
type BTreeLeaf<'Key, 'Value> =  
    class  
        inherit BTreeLeafBase<'Key, 'Value>  
    end
```

Type Parameters

Key



Type of keys used

Value



Type of Values used

The BTreeLeaf(Key, Value) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides BTreeLeafBase(Key, Value).Cache.) |
|  | ObjectsPerPage | (Overrides BTreeLeafBase(Key, Value).ObjectsPerPage.) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |



See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeaf(Key, Value).BTreeLeaf(Key, Value) Properties

The [BTreeLeaf\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides BTreeLeafBase(Key, Value).Cache.) |
|  | ObjectsPerPage | (Overrides BTreeLeafBase(Key, Value).ObjectsPerPage.) |

See Also

[BTreeLeaf\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeaf(Key, Value).Cache Property

By default, the an object cache is determined by a [SessionBase](#) constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[BTreeLeaf\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeaf(Key, Value).ObjectsPerPage Property

[Missing <summary> documentation for "P:VelocityDb.Collection.BTree.BTreeLeaf`2.ObjectsPerPage"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also



[BTreeLeaf\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeaf(Key, Value).BTreeLeaf(Key, Value) Methods

The [BTreeLeaf\(Key, Value\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[BTreeLeaf\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value) Class

Internal class used with BTreeSet and BTreeMap

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

VelocityDb.Collection.BTree.BTreeLeafBase(Key, Value)

[VelocityDb.Collection.BTree.BTreeLeaf\(Key, Value\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class BTreeLeafBase<Key, Value> : BTreeNode
```

VB

```
<SerializableAttribute>  
Public MustInherit Class BTreeLeafBase (Of Key, Value)  
    Inherits BTreeNode
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class BTreeLeafBase abstract : public BTreeNode
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type BTreeLeafBase<'Key, 'Value> =  
    class  
        inherit BTreeNode  
    end
```

Type Parameters

Key






key type

Value



value type

The BTreeLeafBase(Key, Value) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides OptimizedPersistable.Cache.) |
|  FlushIfPageFull | We don't want to flush other BTreeLeaf pages while looking for a placement page (Overrides OptimizedPersistable.FlushIfPageFull.) |
|  KeysCount | Get the number of keys that are in use for this leaf node. |
|  ObjectsPerPage | (Overrides OptimizedPersistable.ObjectsPerPage.) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |






See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).BTreeLeafBase(Key, Value) Properties

The [BTreeLeafBase\(Key, Value\)](#) generic type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides OptimizedPersistable.Cache.) |
|  FlushIfPageFull | We don't want to flush other BTreeLeaf pages while looking for a placement page (Overrides OptimizedPersistable.FlushIfPageFull.) |
|  KeysCount | Get the number of keys that are in use for this leaf node. |
|  ObjectsPerPage | (Overrides OptimizedPersistable.ObjectsPerPage.) |

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).AllowOtherTypesOnSamePage Property

[Missing <summary> documentation for

"P:VelocityDb.Collection.BTree.BTreeLeafBase`2.AllowOtherTypesOnSamePage"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).Cache Property

By default, the an object cache is determined by a [SessionBase](#) constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).FlushIfPageFull Property

We don't want to flush other BTreeLeaf pages while looking for a placement page

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool FlushIfPageFull { get; }
```

VB

```
Public Overrides ReadOnly Property FlushIfPageFull As Boolean  
    Get
```

C++

```
public:  
virtual property bool FlushIfPageFull {  
    bool get () override;  
}
```

F#

```
abstract FlushIfPageFull : bool with get  
override FlushIfPageFull : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.FlushIfPageFull](#)

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).KeysCount Property

Get the number of keys that are in use for this leaf node.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort KeysCount { get; }
```

VB

```
Public ReadOnly Property KeysCount As UShort  
    Get
```

C++

```
public:  
property unsigned short KeysCount {  
    unsigned short get ();  
}
```

F#

```
member KeysCount : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).ObjectsPerPage Property

[Missing <summary> documentation for
"P:VelocityDb.Collection.BTree.BTreeLeafBase`2.ObjectsPerPage"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also



[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeLeafBase(Key, Value).BTreeLeafBase(Key, Value) Methods

The [BTreeLeafBase\(Key, Value\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[BTreeLeafBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value) Class

Represents a collection of keys that is maintained in sorted order. Each key has an associated value. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time. Exceptions are ValueType keys and values.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeBase\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeMapBase\(Key, Value\)](#)

VelocityDb.Collection.BTree.BTreeMap(Key, Value)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class BTreeMap<Key, Value> : BTreeMapBase<Key, Value>,
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>,
    IDictionary<Key, Value>, ICollection<KeyValuePair<Key, Value>>
```

VB

```
<SerializableAttribute>
Public Class BTreeMap(Of Key, Value)
    Inherits BTreeMapBase(Of Key, Value)
    Implements IEnumerable(Of KeyValuePair(Of Key, Value)),
    IEnumerable, IEnumerable(Of Key), IDictionary(Of Key, Value),
    ICollection(Of KeyValuePair(Of Key, Value))
```

C++

```
[SerializableAttribute]
generic<typename Key, typename Value>
public ref class BTreeMap : public BTreeMapBase<Key, Value>,
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>,
    IDictionary<Key, Value>, ICollection<KeyValuePair<Key, Value>>
```

F#

```
[<SerializableAttribute>]
type BTreeMap<'Key, 'Value> =
    class
        inherit BTreeMapBase<'Key, 'Value>
        interface IEnumerable<KeyValuePair<'Key, 'Value>>
```

```

interface IEnumerable
interface IEnumerable<'Key>
interface IDictionary<'Key, 'Value>
interface ICollection<KeyValuePair<'Key, 'Value>>
end

```

Type Parameters

Key


The key type of objects in this BTreeMap

Value




The value type of objects in this BTreeMap

The BTreeMap(Key, Value) type exposes the following members.




Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|------------------------|
|  | BTreeMap(Key, Value) | Creates a new BTreeSet |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | Keys | Gets an ICollection(T) containing the keys of the IDictionary(TKey, TValue) . |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? NO it does not. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
|  | Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear().) |
|  | CopyTo | Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array. |
|  | InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

VelocityDB Class Library

See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value) Constructor

Creates a new BTreeSet

Namespace: [VelocityDb.Collection.BTree](#)**Assembly:** VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeMap (
    VelocityDbComparer<Key> comparer,
    SessionBase session,
    ushort maxEntriesPerNode = 10000,
    ushort comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

VB

```
Public Sub New (
    comparer As VelocityDbComparer(Of Key),
    session As SessionBase,
    Optional maxEntriesPerNode As UShort = 10000,
    Optional comparisonArraySize As UShort = 0,
    Optional comparisonArrayIsCompleteKey As Boolean = false
)
```

C++

```
public:
BTreeMap (
    VelocityDbComparer<Key>^ comparer,
    SessionBase^ session,
    unsigned short maxEntriesPerNode = 10000,
    unsigned short comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

F#

```
new :
    comparer : VelocityDbComparer<'Key> *
    session : SessionBase *
    ?maxEntriesPerNode : uint16 *
    ?comparisonArraySize : uint16 *
    ?comparisonArrayIsCompleteKey : bool
(* Defaults:
    let_maxEntriesPerNode = defaultArg maxEntriesPerNode 10000
    let_comparisonArraySize = defaultArg comparisonArraySize 0
    let_comparisonArrayIsCompleteKey = defaultArg
comparisonArrayIsCompleteKey false
*)
-> BTreeMap
```

VelocityDB Class Library

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

An object comparer. Try using CompareByField

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

maxEntriesPerNode (Optional)

Type: [System.UInt16](#)

Determines internal array and Page sizes

comparisonArraySize (Optional)

Type: [System.UInt16](#)

Determine how many bytes to reserve for each object within a BTree node as a way to avoid actual object compares.

comparisonArrayIsCompleteKey (Optional)

Type: [System.Boolean](#)

If the comparison array bytes are all that needs to be compared to determine ordering then set this to true; otherwise false

See Also




[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).BTreeMap(Key, Value) Properties

The [BTreeMap\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | Keys | Gets an ICollection(T) containing the keys of the IDictionary(TKey, TValue) . |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? NO it does not. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).Comparer Property

Gets the VelocityDbComparer object that is used to compare Key objects.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).Keys Property

Gets an [ICollection\(T\)](#) containing the keys of the [IDictionary\(TKey, TValue\)](#).

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ICollection<Key> Keys { get; }
```

VB

```
Public ReadOnly Property Keys As ICollection(Of Key)  
    Get
```

C++

```
public:  
virtual property ICollection<Key>^ Keys {  
    ICollection<Key>^ get () sealed;  
}
```

F#

```
abstract Keys : ICollection<'Key> with get  
override Keys : ICollection<'Key> with get
```

Return Value

Type: [ICollection\(Key\)](#)

An [ICollection\(T\)](#) containing the keys of the object that implements [IDictionary\(TKey, TValue\)](#).

Implements

[IDictionary\(TKey, TValue\).Keys](#)

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).UsesOidShort Property

Does this BTree use short object references (32 bit) for its internal references? NO it does not.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UsesOidShort { get; }
```

VB

```
Public Overrides ReadOnly Property UsesOidShort As Boolean  
    Get
```

C++

```
public:  
virtual property bool UsesOidShort {  
    bool get () override;  
}
```

F#

```
abstract UsesOidShort : bool with get  
override UsesOidShort : bool with get
```

Property Value

Type: [Boolean](#)

See Also




[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeMap(Key, Value).BTreeMap(Key, Value) Methods

The [BTreeMap\(Key, Value\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
|  Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear() .) |
|  CopyTo | Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array. |
|  InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase) .) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).Clear Method

Removes all elements from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).CopyTo Method

Copies a range of elements from the BTreeBase to a compatible one-dimensional array, starting at the specified index of the target array.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    KeyValuePair<Key, Value>[] array,  
    int arrayIndex,  
    int count  
)
```

VB

```
Public Sub CopyTo (  
    array As KeyValuePair(Of Key, Value) (),  
    arrayIndex As Integer,  
    count As Integer  
)
```

C++

```
public:  
void CopyTo (  
    array<KeyValuePair<Key, Value>>^ array,  
    int arrayIndex,  
    int count  
)
```

F#

```
member CopyTo :  
    array : KeyValuePair<'Key, 'Value>[] *  
    arrayIndex : int *  
    count : int -> unit
```

Parameters

array

Type: [System.Collections.Generic.KeyValuePair\(Key, Value\)](#)[]

The one-dimensional Array that is the destination of the elements copied from BTreeBase. The Array must have zero-based indexing.

arrayIndex

Type: [System.Int32](#)

The zero-based index in the array at which copying begins.

count

VelocityDB Class Library

Type: [System.Int32](#)

The number of elements to copy.

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMap(Key, Value).InitializeAfterRead Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.BTreeMap`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeMap`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeMap\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value) Class

Represents a collection of objects that is maintained in sorted order.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeBase\(Key, Value\)](#)

VelocityDb.Collection.BTree.BTreeMapBase(Key, Value)

[VelocityDb.Collection.BTree.BTreeMap\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeMapOidShort\(Key, Value\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class BTreeMapBase<Key, Value> : BTreeBase<Key, Value>, IEnumer-  
able<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>
```

VB

```
<SerializableAttribute>  
Public MustInherit Class BTreeMapBase(Of Key, Value)  
    Inherits BTreeBase(Of Key, Value)  
    Implements IEnumerable(Of KeyValuePair(Of Key, Value)),  
    IEnumerable, IEnumerable(Of Key)
```

C++

```
[SerializableAttribute]  
generic<typename Key, typename Value>  
public ref class BTreeMapBase abstract : public BTreeBase<Key, Value>, IEnumer-  
able<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type BTreeMapBase<'Key, 'Value> =  
    class  
        inherit BTreeBase<'Key, 'Value>  
        interface IEnumerable<KeyValuePair<'Key, 'Value>>  
        interface IEnumerable  
        interface IEnumerable<'Key>  
    end
```


*Type Parameters***Key**





The type of key objects in this BTree

Value









The type of value objects in a BTreeMap or BTreeMapOidShort

The BTreeMapBase(Key, Value) type exposes the following members.




Properties

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Item | Given a lookup key, returns the corresponding value within the map or null if not found |
|  TransientBatchSize | Get/Set the max batch size used in API such as AddFast(Key) (Overrides BTreeBase(Key, Value).TransientBatchSize.) |
|  ValuePlacement | Set the value placement to be used for all value objects added to this. This setting is not persisted, it is mainly to be used with objects added with AddFast(Key, Value) |
|  Values | Gets a collection containing the values of this object |



Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddFast | Adds an element to an array of to be added objects. The objects in this array are added when a call to FlushTransients() is triggered by multiple events such as calling Count and when collection is committed and/or flushed to disk or when the array is full. The array size is by default currently <code>[!:BTreeBase<Key, Value>.transientBatchBufferDefaultSize]</code> but you can get/set the desired size with TransientBatchSize |
|  EnumerateValues | Enumerates all contained Value objects in corresponding key sorted order |
|  FlushTransients | Adds all queued up to be added objects after presorting them transiently (Overrides BTreeBase(Key, Value).FlushTransients().) |
|  GetEnumerator | Enumerates all contained Key objects in sorted order |
|  GetValueId(Key) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |
|  GetValueId(Key,Byte[]) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |
|  Iterator | Initializes an iterator to find the keys and values of this map |
|  Last | Returns the last KeyValue object in this set. |

VelocityDB Class Library

| | |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
|  LastValue | Returns the last value in this map. |
|  TryGetValue(Key, Value) | Gets the value associated with the specified key. |
|  TryGetValue(Key,Byte[], Value) | Gets the value associated with the specified key. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |





See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).BTreeMapBase(Key, Value) Properties

The [BTreeMapBase\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Item | Given a lookup key, returns the corresponding value within the map or null if not found |
|  | TransientBatchSize | Get/Set the max batch size used in API such as AddFast(Key) (Overrides BTreeBase(Key, Value).TransientBatchSize.) |
|  | ValuePlacement | Set the value placement to be used for all value objects added to this. This setting is not persisted, it is mainly to be used with objects added with AddFast(Key, Value) |
|  | Values | Gets a collection containing the values of this object |

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).Item Property

Given a lookup key, returns the corresponding value within the map or null if not found

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Value this[
    Key key
] { get; set; }
```

VB

```
Public Default Property Item (
    key As Key
) As Value
    Get
    Set
```

C++

```
public:
    virtual property Value default[Key key] {
        Value get (Key key) sealed;
        void set (Key key, Value value) sealed;
    }
```

F#

```
abstract Item : 'Value with get, set
override Item : 'Value with get, set
```

Parameters

key

Type: *Key*

The lookup key

Return Value

Type: *Value*

The value matching the lookup key or null if not found

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).TransientBatchSize Property

Get/Set the max batch size used in API such as [AddFast\(Key\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint TransientBatchSize { get; set; }
```

VB

```
Public Overrides Property TransientBatchSize As UInteger  
    Get  
    Set
```

C++

```
public:  
virtual property unsigned int TransientBatchSize {  
    unsigned int get () override;  
    void set (unsigned int value) override;  
}
```

F#

```
abstract TransientBatchSize : uint32 with get, set  
override TransientBatchSize : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).ValuePlacement Property

Set the value placement to be used for all value objects added to this. This setting is not persisted, it is mainly to be used with objects added with [AddFast\(Key, Value\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Placement ValuePlacement { get; set; }
```

VB

```
Public Property ValuePlacement As Placement  
    Get  
    Set
```

C++

```
public:  
property Placement^ ValuePlacement {  
    Placement^ get ();  
    void set (Placement^ value);  
}
```

F#

```
member ValuePlacement : Placement with get, set
```

Property Value

Type: [Placement](#)

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).Values Property

Gets a collection containing the values of this object

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ICollection<Value> Values { get; }
```

VB

```
Public ReadOnly Property Values As ICollection(Of Value)  
    Get
```

C++

```
public:  
virtual property ICollection<Value>^ Values {  
    ICollection<Value>^ get () sealed;  
}
```

F#

```
abstract Values : ICollection<'Value> with get  
override Values : ICollection<'Value> with get
```

Property Value

Type: [ICollection\(Value\)](#)

See Also












[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeMapBase(Key, Value).BTreeMapBase(Key, Value) Methods

The [BTreeMapBase\(Key, Value\)](#) generic type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddFast | Adds an element to an array of to be added objects. The objects in this array are added when a call to FlushTransients() is triggered by multiple events such as calling Count and when collection is committed and/or flushed to disk or when the array is full. The array size is by default currently <code>[!:BTreeBase<Key, Value>.transientBatchBufferSize]</code> but you can get/set the desired size with TransientBatchSize |
|  EnumerateValues | Enumerates all contained Value objects in corresponding key sorted order |
|  FlushTransients | Adds all queued up to be added objects after presorting them transiently (Overrides BTreeBase(Key, Value).FlushTransients().) |
|  GetEnumerator | Enumerates all contained Key objects in sorted order |
|  GetValueId(Key) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |
|  GetValueId(Key,Byte[]) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |
|  Iterator | Initializes an iterator to find the keys and values of this map |
|  Last | Returns the last KeyValue object in this set. |
|  LastValue | Returns the last value in this map. |
|  TryGetValue(Key, Value) | Gets the value associated with the specified key. |
|  TryGetValue(Key,Byte[], Value) | Gets the value associated with the specified key. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).AddFast Method

Adds an element to an array of to be added objects. The objects in this array are added when a call to [FlushTransients\(\)](#) is triggered by multiple events such as calling [Count](#) and when collection is committed and/or flushed to disk or when the array is full. The array size is by default currently **[!:BTreeBase<Key, Value>.transientBatchBufferDefaultSize]** but you can get/set the desired size with [TransientBatchSize](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddFast (  
    Key key,  
    Value value  
)
```

VB

```
Public Sub AddFast (  
    key As Key,  
    value As Value  
)
```

C++

```
public:  
void AddFast (  
    Key key,  
    Value value  
)
```

F#

```
member AddFast :  
    key : 'Key *  
    value : 'Value -> unit
```

Parameters

key

Type: *Key*

The object beeing added

value

Type: *Value*

The associated value added

VelocityDB Class Library

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).EnumerateValues Method

Enumerates all contained Value objects in corresponding key sorted order

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Value> EnumerateValues ()
```

VB

```
Public Function EnumerateValues As IEnumerable(Of Value)
```

C++

```
public:  
IEnumerable<Value>^ EnumerateValues ()
```

F#

```
member EnumerateValues : unit -> IEnumerable<'Value>
```

Return Value

Type: [IEnumerable\(Value\)](#)

The enumeration

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).FlushTransients Method

Adds all queued up to be added objects after presorting them transiently

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushTransients ()
```

VB

```
Public Overrides Sub FlushTransients
```

C++

```
public:  
virtual void FlushTransients () override
```

F#

```
abstract FlushTransients : unit -> unit  
override FlushTransients : unit -> unit
```

Implements

[IOptimizedPersistable.FlushTransients\(\)](#)

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).GetEnumerator Method

Enumerates all contained Key objects in sorted order

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<KeyValuePair<Key, Value>> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of KeyValuePair(Of Key, Value))
```

C++

```
public:  
IEnumerable<KeyValuePair<Key, Value>>^ GetEnumerator()
```

F#

```
member GetEnumerator : unit -> IEnumerable<KeyValuePair<'Key, 'Value>>
```

Return Value

Type: [IEnumerable\(KeyValuePair\(Key, Value\)\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.BTree.BTreeMapBase`2.GetEnumerator"]



See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).GetValueId Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetValueId(Key) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |
|  | GetValueId(Key,Byte[]) | When Value type implements IOptimizedPersistable , you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired. |

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).GetValueId Method (Key)

When Value type implements [IOptimizedPersistable](#), you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong GetValueId(  
    Key key  
)
```

VB

```
Public Function GetValueId (  
    key As Key  
) As ULong
```

C++

```
public:  
unsigned long long GetValueId(  
    Key key  
)
```

F#

```
member GetValueId :  
    key : 'Key -> uint64
```

Parameters

key

Type: *Key*

Lookup object of type *Key* where Value must be implementing [IOptimizedPersistable](#)

Return Value

Type: [UInt64](#)

The Id of the corresponding value object or 0 if not persistent or not implementing [IOptimizedPersistable](#)

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[GetValueId Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).GetValueId Method (Key, Byte[])

When Value type implements [IOptimizedPersistable](#), you can use this function to get the Id of the persistent object instead of the entire object. Use for performance reasons in certain cases where reading the object isn't desired.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong GetValueId(  
    Key key,  
    byte[] comparisonArray  
)
```

VB

```
Public Function GetValueId (  
    key As Key,  
    comparisonArray As Byte()  
) As ULong
```

C++

```
public:  
unsigned long long GetValueId(  
    Key key,  
    array<unsigned char>^ comparisonArray  
)
```

F#

```
member GetValueId :  
    key : 'Key *  
    comparisonArray : byte[] -> uint64
```

Parameters

key

Type: *Key*

Lookup object of type *Key* where Value must be implementing [IOptimizedPersistable](#)

comparisonArray

Type: [System.Byte\[\]](#)

A byte containing bytes to be compared with.

Return Value

Type: [UInt64](#)

The Id of the corresponding value object or 0 if not persistent or not implementing [IOptimizedPersistable](#)

VelocityDB Class Library

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[GetValueId Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).Iterator Method

Initializes an iterator to find the keys and values of this map

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeMapIterator<Key, Value> Iterator()
```

VB

```
Public Function Iterator As BTreeMapIterator(Of Key, Value)
```

C++

```
public:  
BTreeMapIterator<Key, Value>^ Iterator()
```

F#

```
member Iterator : unit -> BTreeMapIterator<'Key, 'Value>
```

Return Value

Type: [BTreeMapIterator\(Key, Value\)](#)

An iterator

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).Last Method

Returns the last KeyValue object in this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public KeyValuePair<Key, Value> Last ()
```

VB

```
Public Function Last As KeyValuePair(Of Key, Value)
```

C++

```
public:  
KeyValuePair<Key, Value> Last ()
```

F#

```
member Last : unit -> KeyValuePair<'Key, 'Value>
```

Return Value

Type: [KeyValuePair\(Key, Value\)](#)

The last KeyValue in the collection according to the sort order used

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).LastValue Method

Returns the last value in this map.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Value LastValue()
```

VB

```
Public Function LastValue As Value
```

C++

```
public:  
Value LastValue()
```

F#

```
member LastValue : unit -> 'Value
```

Return Value

Type: *Value*

The last value in the collection according to the sort order used



See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).TryGetValue Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------|---------------------------------------------------|
|  | TryGetValue(Key, Value) | Gets the value associated with the specified key. |
|  | TryGetValue(Key,Byte[], Value) | Gets the value associated with the specified key. |

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).TryGetValue Method (Key, Value)

Gets the value associated with the specified key.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetValue(  
    Key key,  
    out Value value  
)
```

VB

```
Public Function TryGetValue (  
    key As Key,  
    <OutAttribute> ByRef value As Value  
) As Boolean
```

C++

```
public:  
virtual bool TryGetValue(  
    Key key,  
    [OutAttribute] Value% value  
) sealed
```

F#

```
abstract TryGetValue :  
    key : 'Key *  
    value : 'Value byref -> bool  
override TryGetValue :  
    key : 'Key *  
    value : 'Value byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

value

Type: *Value*

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the set contains an element with the specified key; otherwise, false.

VelocityDB Class Library

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[TryGetValue Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapBase(Key, Value).TryGetValue Method (Key, Byte[], Value)

Gets the value associated with the specified key.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TryGetValue(  
    Key key,  
    byte[] comparisonArray,  
    out Value value  
)
```

VB

```
Public Function TryGetValue (  
    key As Key,  
    comparisonArray As Byte(),  
    <OutAttribute> ByRef value As Value  
) As Boolean
```

C++

```
public:  
bool TryGetValue(  
    Key key,  
    array<unsigned char>^ comparisonArray,  
    [OutAttribute] Value% value  
)
```

F#

```
member TryGetValue :  
    key : 'Key *  
    comparisonArray : byte[] *  
    value : 'Value byref -> bool
```

Parameters

key

Type: *Key*

The key of the value to get.

comparisonArray

Type: [System.Byte\[\]](#)

The comparison array associated with the key to get.

value

Type: *Value*

VelocityDB Class Library

When this method returns, contains the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Return Value

Type: [Boolean](#)

true if the BTreeBase(Of Key) contains an element with the specified key; otherwise, false.

See Also

[BTreeMapBase\(Key, Value\)Class](#)

[TryGetValue Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value) Class

Iterates all the elements of a BTreeSet

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Collection.BTree.BTreeMapIterator(Key, Value)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class BTreeMapIterator<Key, Value>
```

VB

```
Public Class BTreeMapIterator(Of Key, Value)
```

C++

```
generic<typename Key, typename Value>  
public ref class BTreeMapIterator
```

F#

```
type BTreeMapIterator<'Key, 'Value> = class end
```

Type Parameters

Key


The key type of this collection

Value



The value type of this collection










The BTreeMapIterator(Key, Value) type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------|
|  | IndexInTree | Gets the index of the current iterator element in the iterated collection |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|--------------------------------------------------|
|  | Current | The iterator current Key object |
|  | CurrentKey | Returns the Key at the current iterator position |

| | |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  CurrentValue | Returns the Value at the current iterator position |
|  GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |
|  GoToLast | Positions iterator at the last object in the set. |
|  MoveNext | Advances the enumerator to the next element of the collection. |
|  MovePrevious | Advances the enumerator to the previous element of the map. |
|  Next | Positions the iterator at the next KeyValuePair in the map. Only use this one with nullable Key type. |
|  Previous | Positions the iterator at the previous Key in the set |
|  ReplaceValue | Replaces a value in map |


See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).BTreeMapIterator(Key, Value) Properties

The [BTreeMapIterator\(Key, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------|
|  | IndexInTree | Gets the index of the current iterator element in the iterated collection |

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).IndexInTree Property

Gets the index of the current iterator element in the iterated collection

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long IndexInTree { get; }
```

VB

```
Public ReadOnly Property IndexInTree As Long  
    Get
```

C++

```
public:  
property long long IndexInTree {  
    long long get ();  
}
```

F#

```
member IndexInTree : int64 with get
```

Property Value

Type: [Int64](#)

See Also












[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).BTreeMapIterator(Key, Value) Methods

The [BTreeMapIterator\(Key, Value\)](#) generic type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  Current | The iterator current Key object |
|  CurrentKey | Returns the Key at the current iterator position |
|  CurrentValue | Returns the Value at the current iterator position |
|  GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |
|  GoToLast | Positions iterator at the last object in the set. |
|  MoveNext | Advances the enumerator to the next element of the collection. |
|  MovePrevious | Advances the enumerator to the previous element of the map. |
|  Next | Positions the iterator at the next KeyValuePair in the map. Only use this one with nullable Key type. |
|  Previous | Positions the iterator at the previous Key in the set |
|  ReplaceValue | Replaces a value in map |

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).Current Method

The iterator current Key object

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public KeyValuePair<Key, Value> Current ()
```

VB

```
Public Function Current As KeyValuePair(Of Key, Value)
```

C++

```
public:  
KeyValuePair<Key, Value> Current ()
```

F#

```
member Current : unit -> KeyValuePair<'Key, 'Value>
```

Return Value

Type: [KeyValuePair\(Key, Value\)](#)

The Ket at the current iterator position

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).CurrentKey Method

Returns the Key at the current iterator position

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key CurrentKey()
```

VB

```
Public Function CurrentKey As Key
```

C++

```
public:  
Key CurrentKey()
```

F#

```
member CurrentKey : unit -> 'Key
```

Return Value

Type: *Key*

Key at current position

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).CurrentValue Method

Returns the Value at the current iterator position

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Value CurrentValue ()
```

VB

```
Public Function CurrentValue As Value
```

C++

```
public:  
Value CurrentValue ()
```

F#

```
member CurrentValue : unit -> 'Value
```

Return Value

Type: *Value*

Value at current position



See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).GoTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  | GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).GoTo Method (Key)

Positions iterator at Key object or where Key would be inserted if added to the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GoTo(  
    Key key  
)
```

VB

```
Public Function GoTo (  
    key As Key  
) As Boolean
```

C++

```
public:  
bool GoTo(  
    Key key  
)
```

F#

```
member GoTo :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

The object to position iterator near

Return Value

Type: [Boolean](#)

true if an exact match was found; otherwise false

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[GoTo Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).GoTo Method (Key, Byte[])

Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GoTo(  
    Key key,  
    byte[] comparisonArray  
)
```

VB

```
Public Function GoTo (  
    key As Key,  
    comparisonArray As Byte()  
) As Boolean
```

C++

```
public:  
bool GoTo(  
    Key key,  
    array<unsigned char>^ comparisonArray  
)
```

F#

```
member GoTo :  
    key : 'Key *  
    comparisonArray : byte[] -> bool
```

Parameters

key

Type: *Key*

The object to position iterator near

comparisonArray

Type: [System.Byte\[\]](#)

Contains bytes used for comparing with other arrays for NodeKeys within the set

Return Value

Type: [Boolean](#)

true if an exact match was found; otherwise false

VelocityDB Class Library

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[GoTo Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).GoToLast Method

Positions iterator at the last object in the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void GoToLast ()
```

VB

```
Public Sub GoToLast
```

C++

```
public:  
void GoToLast ()
```

F#

```
member GoToLast : unit -> unit
```

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).MoveNext Method

Advances the enumerator to the next element of the collection.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool MoveNext ()
```

VB

```
Public Function MoveNext As Boolean
```

C++

```
public:  
bool MoveNext ()
```

F#

```
member MoveNext : unit -> bool
```

Return Value

Type: [Boolean](#)

true if the enumerator was successfully advanced to the next element; false if the enumerator has passed the end of the collection.

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).MovePrevious Method

Advances the enumerator to the previous element of the map.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool MovePrevious()
```

VB

```
Public Function MovePrevious As Boolean
```

C++

```
public:  
bool MovePrevious()
```

F#

```
member MovePrevious : unit -> bool
```

Return Value

Type: [Boolean](#)

true if the enumerator was successfully advanced to the previous element; false if the enumerator has reached the position prior to the start of the map.

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).Next Method

Positions the iterator at the next KeyValuePair in the map. Only use this one with nullable Key type.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public KeyValuePair<Key, Value> Next ()
```

VB

```
Public Function Next As KeyValuePair(Of Key, Value)
```

C++

```
public:  
KeyValuePair<Key, Value> Next ()
```

F#

```
member Next : unit -> KeyValuePair<'Key, 'Value>
```

Return Value

Type: [KeyValuePair\(Key, Value\)](#)

The next Key or null if we iterated past the the end of the set.

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).Previous Method

Positions the iterator at the previous Key in the set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public KeyValuePair<Key, Value> Previous()
```

VB

```
Public Function Previous As KeyValuePair(Of Key, Value)
```

C++

```
public:  
KeyValuePair<Key, Value> Previous()
```

F#

```
member Previous : unit -> KeyValuePair<'Key, 'Value>
```

Return Value

Type: [KeyValuePair\(Key, Value\)](#)

The previous Key or null if we iterated to a position before the first Key in the set.

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapIterator(Key, Value).ReplaceValue Method

Replaces a value in map

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ReplaceValue(  
    ref Value value  
)
```

VB

```
Public Sub ReplaceValue (  
    ByRef value As Value  
)
```

C++

```
public:  
void ReplaceValue(  
    Value% value  
)
```

F#

```
member ReplaceValue :  
    value : 'Value byref -> unit
```

Parameters

value

Type: *Value*

the value to replace with

See Also

[BTreeMapIterator\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value) Class

Represents a collection of objects that is maintained in sorted order. Collection and all objects must be within a single Database (since references uses OidShort persistently)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeBase\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeMapBase\(Key, Value\)](#)

VelocityDb.Collection.BTree.BTreeMapOidShort(Key, Value)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class BTreeMapOidShort<Key, Value> : BTreeMapBase<Key, Value>,
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>
```

VB

```
<SerializableAttribute>
Public Class BTreeMapOidShort (Of Key, Value)
    Inherits BTreeMapBase (Of Key, Value)
    Implements IEnumerable (Of KeyValuePair (Of Key, Value)),
        IEnumerable, IEnumerable (Of Key)
```

C++

```
[SerializableAttribute]
generic<typename Key, typename Value>
public ref class BTreeMapOidShort : public BTreeMapBase<Key, Value>,
    IEnumerable<KeyValuePair<Key, Value>>, IEnumerable, IEnumerable<Key>
```

F#

```
[<SerializableAttribute>]
type BTreeMapOidShort<'Key, 'Value> =
    class
        inherit BTreeMapBase<'Key, 'Value>
        interface IEnumerable<KeyValuePair<'Key, 'Value>>
        interface IEnumerable
        interface IEnumerable<'Key>
    end
```

Type Parameters

Key


The object type of keys in this BTreeMapOidShort/>

Value




The value type of values in this BTreeMapOidShort

The BTreeMapOidShort(Key, Value) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|--------------------------------|
|  | BTreeMapOidShort(Key, Value) | Creates a new BTreeMapOidShort |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | MaxNumberOfDatabases | Allow only a single database for this short id (32bit) references collection (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? YES it does. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------|
|  | Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear().) |
|  | InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value) Constructor

Creates a new BTreeMapOidShort

Namespace: [VelocityDb.Collection.BTree](#)**Assembly:** VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeMapOidShort (
    VelocityDbComparer<Key> comparer,
    SessionBase session,
    ushort maxEntriesPerNode = 10000,
    ushort comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

VB

```
Public Sub New (
    comparer As VelocityDbComparer(Of Key),
    session As SessionBase,
    Optional maxEntriesPerNode As UShort = 10000,
    Optional comparisonArraySize As UShort = 0,
    Optional comparisonArrayIsCompleteKey As Boolean = false
)
```

C++

```
public:
BTreeMapOidShort (
    VelocityDbComparer<Key>^ comparer,
    SessionBase^ session,
    unsigned short maxEntriesPerNode = 10000,
    unsigned short comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

F#

```
new :
    comparer : VelocityDbComparer<'Key> *
    session : SessionBase *
    ?maxEntriesPerNode : uint16 *
    ?comparisonArraySize : uint16 *
    ?comparisonArrayIsCompleteKey : bool
(* Defaults:
    let_maxEntriesPerNode = defaultArg maxEntriesPerNode 10000
    let_comparisonArraySize = defaultArg comparisonArraySize 0
    let_comparisonArrayIsCompleteKey = defaultArg
comparisonArrayIsCompleteKey false
*)
-> BTreeMapOidShort
```

VelocityDB Class Library

Parameters

comparer

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

An object comparer. Try using CompareByField

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

maxEntriesPerNode (Optional)

Type: [System.UInt16](#)

Determines internal array and Page sizes

comparisonArraySize (Optional)

Type: [System.UInt16](#)

Determine how many bytes to reserve for each object within a BTree node as a way to avoid actual object compares.

comparisonArrayIsCompleteKey (Optional)

Type: [System.Boolean](#)

If the comparison array bytes are all that needs to be compared to determine ordering then set this to true; otherwise false

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)




[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).BTreeMapOidShort(Key, Value)

Properties

The [BTreeMapOidShort\(Key, Value\)](#) generic type exposes the following members.

Properties

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  MaxNumberOfDatabases | Allow only a single database for this short id (32bit) references collection (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? YES it does. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).Comparer Property

Gets the VelocityDbComparer object that is used to compare Key objects.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).MaxNumberOfDatabases Property

Allow only a single database for this short id (32bit) references collection

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).UsesOidShort Property

Does this BTree use short object references (32 bit) for its internal references? YES it does.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UsesOidShort { get; }
```

VB

```
Public Overrides ReadOnly Property UsesOidShort As Boolean  
    Get
```

C++

```
public:  
virtual property bool UsesOidShort {  
    bool get () override;  
}
```

F#

```
abstract UsesOidShort : bool with get  
override UsesOidShort : bool with get
```

Property Value

Type: [Boolean](#)

See Also



[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeMapOidShort(Key, Value).BTreeMapOidShort(Key, Value) Methods

The [BTreeMapOidShort\(Key, Value\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------|
|  | Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear() .) |
|  | InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).Clear Method

Removes all elements from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeMapOidShort(Key, Value).InitializeAfterRead Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.BTreeMapOidShort`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeMapOidShort`2.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeMapOidShort\(Key, Value\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeNode Class

A BTree consists of a tree of nodes. Each BTree node has this class a base class.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Collection.BTree.BTreeNode

[VelocityDb.Collection.BTree.BTreeBase\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeInternalBase\(Key, Value\)](#)

[VelocityDb.Collection.BTree.BTreeLeafBase\(Key, Value\)](#)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class BTreeNode : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public MustInherit Class BTreeNode  
    Inherits OptimizedPersistable
```

C++

```
[SerializableAttribute]  
public ref class BTreeNode abstract : public OptimizedPersistable
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type BTreeNode =  
    class  
        inherit OptimizedPersistable  
    end
```

See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key) Class

Represents a collection of objects that is maintained in sorted order. A persistent BTree references its contained objects by Oid instead of direct object references. This way, we will only open the referenced objects on demand which reduces memory usage and initial BTree load time.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

VelocityDb.Collection.BTree.BTreeSet(Key)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class BTreeSet<Key> : BTreeBase<Key, Key>,
    ISet<Key>, ICollection<Key>, IEnumerable<Key>, IEnumerable
```

VB

```
<SerializableAttribute>
Public Class BTreeSet (Of Key)
    Inherits BTreeBase (Of Key, Key)
    Implements ISet (Of Key), ICollection (Of Key),
    IEnumerable (Of Key), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename Key>
public ref class BTreeSet : public BTreeBase<Key, Key>,
    ISet<Key>, ICollection<Key>, IEnumerable<Key>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type BTreeSet<'Key> =
    class
        inherit BTreeBase<'Key, 'Key>
        interface ISet<'Key>
        interface ICollection<'Key>
        interface IEnumerable<'Key>
        interface IEnumerable
    end
```


Type Parameters

Key



The key type of objects in this BTree

The BTreeSet(Key) type exposes the following members.






Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|------------------------|
|  | BTreeSet(Key) | Creates a new BTreeSet |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? NO it does not. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear().) |
|  | First | Returns the first Key object in this set. |
|  | InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase).) |
|  | Iterator | Initializes an iterator to find the keys of this set (Overrides BTreeBase(Key, Value).Iterator().) |
|  | Last | Returns the last Key object in this set. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key) Constructor

Creates a new BTreeSet

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet (
    VelocityDbComparer<Key> comparer = null,
    SessionBase session = null,
    ushort maxEntriesPerNode = 10000,
    ushort comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

VB

```
Public Sub New (
    Optional comparer As VelocityDbComparer(Of Key) = Nothing,
    Optional session As SessionBase = Nothing,
    Optional maxEntriesPerNode As UShort = 10000,
    Optional comparisonArraySize As UShort = 0,
    Optional comparisonArrayIsCompleteKey As Boolean = false
)
```

C++

```
public:
    BTreeSet (
        VelocityDbComparer<Key>^ comparer = nullptr,
        SessionBase^ session = nullptr,
        unsigned short maxEntriesPerNode = 10000,
        unsigned short comparisonArraySize = 0,
        bool comparisonArrayIsCompleteKey = false
    )
```

F#

```
new :
    ?comparer : VelocityDbComparer<'Key> *
    ?session : SessionBase *
    ?maxEntriesPerNode : uint16 *
    ?comparisonArraySize : uint16 *
    ?comparisonArrayIsCompleteKey : bool
(* Defaults:
    let_comparer = defaultArg comparer null
    let_session = defaultArg session null
    let_maxEntriesPerNode = defaultArg maxEntriesPerNode 10000
    let_comparisonArraySize = defaultArg comparisonArraySize 0
    let_comparisonArrayIsCompleteKey = defaultArg
comparisonArrayIsCompleteKey false
```

```
*)  
-> BTreeSet
```

Parameters

comparer (Optional)

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

An object comparer. Try using CompareByField

session (Optional)

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

maxEntriesPerNode (Optional)

Type: [System.UInt16](#)

Determines internal array and Page sizes

comparisonArraySize (Optional)

Type: [System.UInt16](#)

Determine how many bytes to reserve for each object within a BTree node as a way to avoid actual object compares.

comparisonArrayIsCompleteKey (Optional)

Type: [System.Boolean](#)

If the comparison array bytes are all that needs to be compared to determine ordering then set this to true; otherwise false

See Also



[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).BTreeSet(Key) Properties

The [BTreeSet\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? NO it does not. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).Comparer Property

Gets the VelocityDbComparer object that is used to compare Key objects.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).UsesOidShort Property

Does this BTree use short object references (32 bit) for its internal references? NO it does not.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UsesOidShort { get; }
```

VB

```
Public Overrides ReadOnly Property UsesOidShort As Boolean  
    Get
```

C++

```
public:  
virtual property bool UsesOidShort {  
    bool get () override;  
}
```

F#

```
abstract UsesOidShort : bool with get  
override UsesOidShort : bool with get
```

Property Value

Type: [Boolean](#)

See Also






[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeSet(Key).BTreeSet(Key) Methods

The [BTreeSet\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear() .) |
|  First | Returns the first Key object in this set. |
|  InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase) .) |
|  Iterator | Initializes an iterator to find the keys of this set (Overrides BTreeBase(Key, Value).Iterator() .) |
|  Last | Returns the last Key object in this set. |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).Clear Method

Removes all elements from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).First Method

Returns the first Key object in this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key First()
```

VB

```
Public Function First As Key
```

C++

```
public:  
Key First()
```

F#

```
member First : unit -> 'Key
```

Return Value

Type: *Key*

The first Key in the collection according to the sort order used

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).InitializeAfterRead Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.BTreeSet`1.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeSet`1.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).Iterator Method

Initializes an iterator to find the keys of this set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override BTreeSetIterator<Key> Iterator ()
```

VB

```
Public Overrides Function Iterator As BTreeSetIterator (Of Key)
```

C++

```
public:  
virtual BTreeSetIterator<Key>^ Iterator () override
```

F#

```
abstract Iterator : unit -> BTreeSetIterator<'Key>  
override Iterator : unit -> BTreeSetIterator<'Key>
```

Return Value

Type: [BTreeSetIterator\(Key\)](#)

iterator

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSet(Key).Last Method

Returns the last Key object in this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key Last ()
```

VB

```
Public Function Last As Key
```

C++

```
public:  
Key Last ()
```

F#

```
member Last : unit -> 'Key
```

Return Value

Type: *Key*

The last Key in the collection according to the sort order used

See Also

[BTreeSet\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key) Class

Iterates all the elements of a BTreeSet

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Collection.BTree.BTreeSetIterator(Key)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class BTreeSetIterator<Key>
```

VB

```
Public Class BTreeSetIterator (Of Key)
```

C++

```
generic<typename Key>  
public ref class BTreeSetIterator
```

F#

```
type BTreeSetIterator<'Key> = class end
```


Type Parameters

Key




The type of the elements contained in the BTreeSet









The BTreeSetIterator(Key) type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------|
|  | IndexInTree | Gets the index of the current iterator element in the iterated collection |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------|-------------------------------------------|
|  | Current | The iterator current Key object |
|  | CurrentComparisonArray | The iterator current Key comparison array |
|  | ElementAt | Gets an item at a certain index |

| | |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |
|  GoToLast | Positions iterator at the last object in the set. |
|  MoveNext | Advances the enumerator to the next element of the collection. |
|  MovePrevious | Advances the enumerator to the previous element of the collection. |
|  Next | Positions the iterator at the next Key in the set. Only use this one with nullable Key type. |
|  Previous | Positions the iterator at the previous Key in the set |
|  Remove | Not yet ready for public use (testing internally first) |


See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).BTreeSetIterator(Key) Properties

The [BTreeSetIterator\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------|
|  | IndexInTree | Gets the index of the current iterator element in the iterated collection |

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).IndexInTree Property

Gets the index of the current iterator element in the iterated collection

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long IndexInTree { get; }
```

VB

```
Public ReadOnly Property IndexInTree As Long  
    Get
```

C++

```
public:  
property long long IndexInTree {  
    long long get ();  
}
```

F#

```
member IndexInTree : int64 with get
```

Property Value

Type: [Int64](#)

See Also












[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).BTreeSetIterator(Key) Methods

The [BTreeSetIterator\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  Current | The iterator current Key object |
|  CurrentComparisonArray | The iterator current Key comparison array |
|  ElementAt | Gets an item at a certain index |
|  GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |
|  GoToLast | Positions iterator at the last object in the set. |
|  MoveNext | Advances the enumerator to the next element of the collection. |
|  MovePrevious | Advances the enumerator to the previous element of the collection. |
|  Next | Positions the iterator at the next Key in the set. Only use this one with nullable Key type. |
|  Previous | Positions the iterator at the previous Key in the set |
|  Remove | Not yet ready for public use (testing internally first) |

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).Current Method

The iterator current Key object

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key Current ()
```

VB

```
Public Function Current As Key
```

C++

```
public:  
Key Current ()
```

F#

```
member Current : unit -> 'Key
```

Return Value

Type: *Key*

The Ket at the current iterator position

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).CurrentComparisonArray Method

The iterator current Key comparison array

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public byte[] CurrentComparisonArray()
```

VB

```
Public Function CurrentComparisonArray As Byte()
```

C++

```
public:  
array<unsigned char>^ CurrentComparisonArray()
```

F#

```
member CurrentComparisonArray : unit -> byte[]
```

Return Value

Type: [Byte\[\]](#)

The Key comparison array at the current iterator position

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).ElementAt Method

Gets an item at a certain index

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key ElementAt (  
    long i  
)
```

VB

```
Public Function ElementAt (  
    i As Long  
) As Key
```

C++

```
public:  
Key ElementAt (  
    long long i  
)
```

F#

```
member ElementAt :  
    i : int64 -> 'Key
```

Parameters

i

Type: [System.Int64](#)

the item index

Return Value

Type: *Key*

the item at the specified index



See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).GoTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GoTo(Key) | Positions iterator at Key object or where Key would be inserted if added to the set. |
|  | GoTo(Key,Byte[]) | Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb. |

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).GoTo Method (Key)

Positions iterator at Key object or where Key would be inserted if added to the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GoTo(  
    Key key  
)
```

VB

```
Public Function GoTo (  
    key As Key  
) As Boolean
```

C++

```
public:  
bool GoTo(  
    Key key  
)
```

F#

```
member GoTo :  
    key : 'Key -> bool
```

Parameters

key

Type: *Key*

The object to position iterator near

Return Value

Type: [Boolean](#)

true if an exact match was found; otherwise false

See Also

[BTreeSetIterator\(Key\)Class](#)

[GoTo Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).GoTo Method (Key, Byte[])

Positions iterator at Key object or where Key would be inserted if added to the set. This function is normally only used internally by VelocityDb.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GoTo(  
    Key key,  
    byte[] comparisonArray  
)
```

VB

```
Public Function GoTo (  
    key As Key,  
    comparisonArray As Byte()  
) As Boolean
```

C++

```
public:  
bool GoTo(  
    Key key,  
    array<unsigned char>^ comparisonArray  
)
```

F#

```
member GoTo :  
    key : 'Key *  
    comparisonArray : byte[] -> bool
```

Parameters

key

Type: *Key*

The object to position iterator near

comparisonArray

Type: [System.Byte\[\]](#)

Contains bytes used for comparing with other arrays for NodeKeys within the set

Return Value

Type: [Boolean](#)

true if an exact match was found; otherwise false

VelocityDB Class Library

See Also

[BTreeSetIterator\(Key\)Class](#)

[GoTo Overload](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).GoToLast Method

Positions iterator at the last object in the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void GoToLast ()
```

VB

```
Public Sub GoToLast
```

C++

```
public:  
void GoToLast ()
```

F#

```
member GoToLast : unit -> unit
```

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).MoveNext Method

Advances the enumerator to the next element of the collection.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool MoveNext ()
```

VB

```
Public Function MoveNext As Boolean
```

C++

```
public:  
bool MoveNext ()
```

F#

```
member MoveNext : unit -> bool
```

Return Value

Type: [Boolean](#)

true if the enumerator was successfully advanced to the next element; false if the enumerator has passed the end of the collection.

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).MovePrevious Method

Advances the enumerator to the previous element of the collection.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool MovePrevious ()
```

VB

```
Public Function MovePrevious As Boolean
```

C++

```
public:  
bool MovePrevious ()
```

F#

```
member MovePrevious : unit -> bool
```

Return Value

Type: [Boolean](#)

true if the enumerator was successfully advanced to the previous element; false if the enumerator has reached the position prior to the the start of the collection.

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).Next Method

Positions the iterator at the next Key in the set. Only use this one with nullable Key type.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key Next ()
```

VB

```
Public Function Next As Key
```

C++

```
public:  
Key Next ()
```

F#

```
member Next : unit -> 'Key
```

Return Value

Type: *Key*

The next Key or null if we iterated past the the end of the set.

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).Previous Method

Positions the iterator at the previous Key in the set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key Previous ()
```

VB

```
Public Function Previous As Key
```

C++

```
public:  
Key Previous ()
```

F#

```
member Previous : unit -> 'Key
```

Return Value

Type: *Key*

The previous Key or null if we iterated to a position before the first Key in the set.

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetIterator(Key).Remove Method

Not yet ready for public use (testing internally first)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove ()
```

VB

```
Public Sub Remove
```

C++

```
public:  
void Remove ()
```

F#

```
member Remove : unit -> unit
```

See Also

[BTreeSetIterator\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key) Class

Represents a collection of objects that is maintained in sorted order. Collection and all objects must be within a single Database (since references uses OidShort persistently)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.BTree.BTreeNode](#)

[VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

VelocityDb.Collection.BTree.BTreeSetOidShort(Key)

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class BTreeSetOidShort<Key> : BTreeBase<Key, Key>,
    IEnumerable<Key>, IEnumerable
```

VB

```
<SerializableAttribute>
Public Class BTreeSetOidShort (Of Key)
    Inherits BTreeBase (Of Key, Key)
    Implements IEnumerable (Of Key), IEnumerable
```

C++

```
[SerializableAttribute]
generic<typename Key>
public ref class BTreeSetOidShort : public BTreeBase<Key, Key>,
    IEnumerable<Key>, IEnumerable
```

F#

```
[<SerializableAttribute>]
type BTreeSetOidShort<'Key> =
    class
        inherit BTreeBase<'Key, 'Key>
        interface IEnumerable<'Key>
        interface IEnumerable
    end
```


Type Parameters

Key




The object type of objects in this BTree, currently must be a subclass of [OptimizedPersistable](#)

The BTreeSetOidShort(Key) type exposes the following members.






Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|--------------------------------|
|  | BTreeSetOidShort(Key) | Creates a new BTreeSetOidShort |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | MaxNumberOfDatabases | Allow only a single database for this short id (32bit) references collection (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? YES it does. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  | Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear().) |
|  | First | Returns the first Key object in this set. |
|  | InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase).) |
|  | Iterator | Initializes an iterator to find the keys of this set (Overrides BTreeBase(Key, Value).Iterator().) |
|  | Last | Returns the last Key object in this set. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key) Constructor

Creates a new BTreeSetOidShort

Namespace: [VelocityDb.Collection.BTree](#)**Assembly:** VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSetOidShort (
    VelocityDbComparer<Key> comparer = null,
    SessionBase session = null,
    ushort maxEntriesPerNode = 10000,
    ushort comparisonArraySize = 0,
    bool comparisonArrayIsCompleteKey = false
)
```

VB

```
Public Sub New (
    Optional comparer As VelocityDbComparer(Of Key) = Nothing,
    Optional session As SessionBase = Nothing,
    Optional maxEntriesPerNode As UShort = 10000,
    Optional comparisonArraySize As UShort = 0,
    Optional comparisonArrayIsCompleteKey As Boolean = false
)
```

C++

```
public:
    BTreeSetOidShort (
        VelocityDbComparer<Key>^ comparer = nullptr,
        SessionBase^ session = nullptr,
        unsigned short maxEntriesPerNode = 10000,
        unsigned short comparisonArraySize = 0,
        bool comparisonArrayIsCompleteKey = false
    )
```

F#

```
new :
    ?comparer : VelocityDbComparer<'Key> *
    ?session : SessionBase *
    ?maxEntriesPerNode : uint16 *
    ?comparisonArraySize : uint16 *
    ?comparisonArrayIsCompleteKey : bool
(* Defaults:
    let _comparer = defaultArg comparer null
    let _session = defaultArg session null
    let _maxEntriesPerNode = defaultArg maxEntriesPerNode 10000
    let _comparisonArraySize = defaultArg comparisonArraySize 0
    let _comparisonArrayIsCompleteKey = defaultArg
comparisonArrayIsCompleteKey false
```

```
*)  
-> BTreeSetOidShort
```

Parameters

comparer (Optional)

Type: [VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

An object comparer. Try using CompareByField

session (Optional)

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

maxEntriesPerNode (Optional)

Type: [System.UInt16](#)

Determines internal array and Page sizes

comparisonArraySize (Optional)

Type: [System.UInt16](#)

Determine how many bytes to reserve for each object within a BTree node as a way to avoid actual object compares.

comparisonArrayIsCompleteKey (Optional)

Type: [System.Boolean](#)

If the comparison array bytes are all that needs to be compared to determine ordering then set this to true; otherwise false

See Also




[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).BTreeSetOidShort(Key) Properties

The [BTreeSetOidShort\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Comparer | Gets the VelocityDbComparer object that is used to compare Key objects. (Overrides BTreeBase(Key, Value).Comparer.) |
|  | MaxNumberOfDatabases | Allow only a single database for this short id (32bit) references collection (Overrides OptimizedPersistable.MaxNumberOfDatabases.) |
|  | UsesOidShort | Does this BTree use short object references (32 bit) for its internal references? YES it does. (Overrides BTreeBase(Key, Value).UsesOidShort.) |

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).Comparer Property

Gets the VelocityDbComparer object that is used to compare Key objects.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override VelocityDbComparer<Key> Comparer { get; }
```

VB

```
Public Overrides ReadOnly Property Comparer As VelocityDbComparer(Of Key)  
    Get
```

C++

```
public:  
virtual property VelocityDbComparer<Key>^ Comparer {  
    VelocityDbComparer<Key>^ get () override;  
}
```

F#

```
abstract Comparer : VelocityDbComparer<'Key> with get  
override Comparer : VelocityDbComparer<'Key> with get
```

Property Value

Type: [VelocityDbComparer\(Key\)](#)

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).MaxNumberOfDatabases Property

Allow only a single database for this short id (32bit) references collection

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint MaxNumberOfDatabases { get; }
```

VB

```
Public Overrides ReadOnly Property MaxNumberOfDatabases As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int MaxNumberOfDatabases {  
    unsigned int get () override;  
}
```

F#

```
abstract MaxNumberOfDatabases : uint32 with get  
override MaxNumberOfDatabases : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.MaxNumberOfDatabases](#)

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).UsesOidShort Property

Does this BTree use short object references (32 bit) for its internal references? YES it does.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UsesOidShort { get; }
```

VB

```
Public Overrides ReadOnly Property UsesOidShort As Boolean  
    Get
```

C++

```
public:  
virtual property bool UsesOidShort {  
    bool get () override;  
}
```

F#

```
abstract UsesOidShort : bool with get  
override UsesOidShort : bool with get
```

Property Value

Type: [Boolean](#)

See Also






[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)



BTreeSetOidShort(Key).BTreeSetOidShort(Key) Methods

The [BTreeSetOidShort\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  Clear | Removes all elements from the set. (Overrides BTreeBase(Key, Value).Clear() .) |
|  First | Returns the first Key object in this set. |
|  InitializeAfterRead | (Overrides BTreeBase(Key, Value).InitializeAfterRead(SessionBase) .) |
|  Iterator | Initializes an iterator to find the keys of this set (Overrides BTreeBase(Key, Value).Iterator() .) |
|  Last | Returns the last Key object in this set. |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).Clear Method

Removes all elements from the set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Clear ()
```

VB

```
Public Overrides Sub Clear
```

C++

```
public:  
virtual void Clear () override
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).First Method

Returns the first Key object in this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key First()
```

VB

```
Public Function First As Key
```

C++

```
public:  
Key First()
```

F#

```
member First : unit -> 'Key
```

Return Value

Type: *Key*

The first Key in the collection according to the sort order used

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).InitializeAfterRead Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.BTree.BTreeSetOidShort`1.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Collection.BTree.BTreeSetOidShort`1.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).Iterator Method

Initializes an iterator to find the keys of this set

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override BTreeSetIterator<Key> Iterator ()
```

VB

```
Public Overrides Function Iterator As BTreeSetIterator(Of Key)
```

C++

```
public:  
virtual BTreeSetIterator<Key>^ Iterator () override
```

F#

```
abstract Iterator : unit -> BTreeSetIterator<'Key>  
override Iterator : unit -> BTreeSetIterator<'Key>
```

Return Value

Type: [BTreeSetIterator\(Key\)](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.BTree.BTreeSetOidShort`1.Iterator"]

See Also

[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

BTreeSetOidShort(Key).Last Method

Returns the last Key object in this set.

Namespace: [VelocityDb.Collection.BTree](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Key Last ()
```

VB

```
Public Function Last As Key
```

C++

```
public:  
Key Last ()
```

F#

```
member Last : unit -> 'Key
```

Return Value

Type: *Key*

The last Key in the collection according to the sort order used

See Also


[BTreeSetOidShort\(Key\)Class](#)

[VelocityDb.Collection.BTree Namespace](#)

VelocityDb.Collection.BTree.Extensions Namespace

The `VelocityDb.Collection.BTree.Extensions` namespace contains a few extensions to improve performance of Linq for Objects queries. The same code can also be found in open source as `VelocityDBExtensions.Extensions.BTree`

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | BTreeExtensions | A few extensions to improve performance of Linq for Objects queries We need your HELP to improve it to cover more use cases of queries! |

BTreeExtensions Class

A few extensions to improve performance of Linq for Objects queries We need your HELP to improve it to cover more use cases of queries!

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Collection.BTree.Extensions.BTreeExtensions

Namespace: [VelocityDb.Collection.BTree.Extensions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class BTreeExtensions
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class BTreeExtensions
```

C++



```
[ExtensionAttribute]  
public ref class BTreeExtensions abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type BTreeExtensions = class end
```

The **BTreeExtensions** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------|
|  | Count(Key) | Override to improve performance over IEnumerable LINQ extension |
|  | Where(Key) | Override to improve performance over IEnumerable LINQ extension |



See Also

[VelocityDb.Collection.BTree.Extensions Namespace](#)

BTreeExtensions.BTreeExtensions Methods

The [BTreeExtensions](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------|
|  | Count(Key) | Override to improve performance over IEnumerable LINQ extension |
|  | Where(Key) | Override to improve performance over IEnumerable LINQ extension |

See Also

[BTreeExtensions Class](#)

[VelocityDb.Collection.BTree.Extensions Namespace](#)

BTreeExtensions.Count(Key) Method

Override to improve performance over IEnumerable LINQ extension

Namespace: [VelocityDb.Collection.BTree.Extensions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int Count<Key>(
    this BTreeBase<Key, Key> sourceCollection
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Count(Of Key) (
    sourceCollection As BTreeBase(Of Key, Key)
) As Integer
```

C++

```
public:
[ExtensionAttribute]
generic<typename Key>
static int Count(
    BTreeBase<Key, Key>^ sourceCollection
)
```

F#

```
[<ExtensionAttribute>]
static member Count :
    sourceCollection : BTreeBase<'Key, 'Key> -> int
```

Parameters

sourceCollection

Type: [VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

the collection

Type Parameters

Key

key type

Return Value

Type: [Int32](#)

Size of the collection

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [BTreeBase](#)(**Key**, **Key**). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[BTreeExtensions Class](#)

[VelocityDb.Collection.BTree.Extensions Namespace](#)

BTreeExtensions.Where(Key) Method

Override to improve performance over IEnumerable LINQ extension

Namespace: [VelocityDb.Collection.BTree.Extensions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEnumerable<Key> Where<Key>(
    this BTreeBase<Key, Key> sourceCollection,
    Expression<Func<Key, bool>> expr
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Where(Of Key) (
    sourceCollection As BTreeBase(Of Key, Key),
    expr As Expression(Of Func(Of Key, Boolean))
) As IEnumerable(Of Key)
```

C++

```
public:
[ExtensionAttribute]
generic<typename Key>
static IEnumerable<Key>^ Where(
    BTreeBase<Key, Key>^ sourceCollection,
    Expression<Func<Key, bool>^>^ expr
)
```

F#

```
[<ExtensionAttribute>]
static member Where :
    sourceCollection : BTreeBase<'Key, 'Key> *
    expr : Expression<Func<'Key, bool>> -> IEnumerable<'Key>
```

Parameters

sourceCollection

Type: [VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

the collection

expr

Type: [System.Linq.Expressions.Expression\(Func\(Key, Boolean\)\)](#)

an expression

Type Parameters

Key

key type

Return Value

Type: [IEnumerable](#)(**Key**)

Enumeration of collection where the expression evaluates to true

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [BTreeBase](#)(**Key, Key**). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also







[BTreeExtensions Class](#)

[VelocityDb.Collection.BTree.Extensions Namespace](#)

VelocityDb.Collection.Comparer Namespace

The `VelocityDb.Collection.Comparer` namespace contains classes for the VelocityDb Comparer classes optimized for persistent storage in VelocityDb databases

Classes

| Class | Description |
|----------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ByteArrayComparer | Compares two byte arrays |
|  CompareByField(Key) | Use this predefined comparator whenever the sorting can be defined by specifying one or more fields of the sorted Key type. |
|  CompareByFieldIndex(Key) | Internal use for Index implementation |
|  HashCodeComparer(T) | When the type of object being compared is a simple type like a string and you don't care about the sort order, then this comparator is what you want to use. It is similar to using a hash based collection but with less overhead (more compact) This kind of comparator especially shines when you create the BTreeSet/BTreeMap using a comparisonByteArray size of 4 |
|  SimpleComparer(T) | When the type of object being compared is a simple type like string, int, ..., this is the comparator you probably want to use. |
|  VelocityDbComparer(Key) | The base class of for comparing persistent objects. If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared |

ByteArrayComparer Class

Compares two byte arrays

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Collection.Comparer.ByteArrayComparer

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class ByteArrayComparer : IComparer<byte[]>
```

VB

```
<SerializableAttribute>
Public Class ByteArrayComparer
    Implements IComparer(Of Byte())
```

C++


```
[SerializableAttribute]
public ref class ByteArrayComparer : IComparer<array<unsigned char>^>
```

F#


```
[<SerializableAttribute>]
type ByteArrayComparer =
    class
        interface IComparer<byte[]>
    end
```

The **ByteArrayComparer** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | ByteArrayComparer | Initializes a new instance of the ByteArrayComparer class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |
|  | memcmp | |

VelocityDB Class Library

See Also

[VelocityDb.Collection.Comparer Namespace](#)

ByteArrayComparer Constructor

Initializes a new instance of the [ByteArrayComparer](#) class

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ByteArrayComparer ()
```

VB

```
Public Sub New
```

C++

```
public:  
ByteArrayComparer ()
```

F#

```
new : unit -> ByteArrayComparer
```

See Also

[ByteArrayComparer Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

ByteArrayComparer.ByteArrayComparer Methods

The [ByteArrayComparer](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |
|  | memcmp | |

See Also

[ByteArrayComparer Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

ByteArrayComparer.Compare Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.Compare(System.Byte[],System.Byte[])"]

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Compare(  
    byte[] a,  
    byte[] b  
)
```

VB

```
Public Function Compare (  
    a As Byte(),  
    b As Byte()  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    array<unsigned char>^ a,  
    array<unsigned char>^ b  
) sealed
```

F#

```
abstract Compare :  
    a : byte[] *  
    b : byte[] -> int  
override Compare :  
    a : byte[] *  
    b : byte[] -> int
```

Parameters

a

Type: [System.Byte\[\]](#)

[Missing <param name="a"/> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.Compare(System.Byte[],System.Byte[])"]

b

Type: [System.Byte\[\]](#)

[Missing <param name="b"/> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.Compare(System.Byte[],System.Byte[])"]

VelocityDB Class Library

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.Compare(System.Byte[],System.Byte[])"]

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[ByteArrayComparer Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

ByteArrayComparer.memcmp Method

[Missing <summary> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.memcmp(System.Byte[],System.Byte[],System.Int32)"]

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int memcmp(  
    byte[] b1,  
    byte[] b2,  
    int count  
)
```

VB

```
Public Shared Function memcmp (  
    b1 As Byte(),  
    b2 As Byte(),  
    count As Integer  
) As Integer
```

C++

```
public:  
static int memcmp(  
    array<unsigned char>^ b1,  
    array<unsigned char>^ b2,  
    int count  
)
```

F#

```
static member memcmp :  
    b1 : byte[] *  
    b2 : byte[] *  
    count : int -> int
```

Parameters

b1

Type: [System.Byte\[\]](#)

[Missing <param name="b1"/> documentation for

"M:VelocityDb.Collection.Comparer.ByteArrayComparer.memcmp(System.Byte[],System.Byte[],System.Int32)"]

b2

Type: [System.Byte\[\]](#)

[Missing <param name="b2"/> documentation for "M:VelocityDb.Collection.Comparer.ByteArrayComparer.memcmp(System.Byte[],System.Byte[],System.Int32)"]

count

Type: [System.Int32](#)

[Missing <param name="count"/> documentation for "M:VelocityDb.Collection.Comparer.ByteArrayComparer.memcmp(System.Byte[],System.Byte[],System.Int32)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.Comparer.ByteArrayComparer.memcmp(System.Byte[],System.Byte[],System.Int32)"]

See Also

[ByteArrayComparer Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key) Class

Use this predefined comparator whenever the sorting can be defined by specifying one or more fields of the sorted Key type.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

VelocityDb.Collection.Comparer.CompareByField(Key)

[VelocityDb.Collection.Comparer.CompareByFieldIndex\(Key\)](#)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class CompareByField<Key> : VelocityDbComparer<Key>
```

VB

```
<SerializableAttribute>  
Public Class CompareByField(Of Key)  
    Inherits VelocityDbComparer(Of Key)
```

C++

```
[SerializableAttribute]  
generic<typename Key>  
public ref class CompareByField : public VelocityDbComparer<Key>
```

F#

```
[<SerializableAttribute>]  
type CompareByField<'Key> =  
    class  
        inherit VelocityDbComparer<'Key>  
    end
```



Type Parameters

Key




The type of object being compared (containing field(s) to compare)

The CompareByField(Key) type exposes the following members.









Constructors


| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareByField(Key)(SessionBase) | This special constructor is provided so that VelocityDb can instantiate the object after reading it from disk. This is also the place to initialize transient variables of the object. |
|  | CompareByField(Key)(String, SessionBase, Boolean, Boolean, Boolean) | Compares objects by comparing one or more data fields of type Key |

Properties



| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddOidCompare | Is the object Oid going to be compared when all the compared fields are equal? (Overrides VelocityDbComparer(Key).AddOidCompare.) |
|  | FieldsToCompare | |
|  | OidPartOfComparisonBytes | Is the Oid (or OidShort) included in the comparison array bytes? (Overrides VelocityDbComparer(Key).OidPartOfComparisonBytes.) |

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddFieldToCompare | Add an additional field to compare when prior fields all are equal |
|  | Compare(Key, Key) | Compares field(s) of Key a with field(s) of Key b (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |
|  | Compare(Byte[], Int32, Byte[], Key) | (Overrides VelocityDbComparer(Key).Compare(Byte[], Int32, Byte[], Key).) |
|  | CompareField | For internal use. Compares a field value in two objects |
|   | GetHashCode32 | Microsoft messed up the 64 bit version of GetHashCode. Use this original 32-bit version to get consistent String hash codes across 32 bit and 64 bit builds. See remarks in https://msdn.microsoft.com/en-us/library/system.string.gethashcode(v=vs.110).aspx |
|  | InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase).) |
|  | SetComparisonArrayFromObject | Sets a byte array equivalent to the field values to compare. Using this bypasses object compares, instead the byte arrays are compared byte by byte until a diff is found. If a byte diff isn't found then object compares may happen. Field types currently supported: byte, bool, Char, string, Int16, Int32, Int64, UInt16, UInt32, UInt64, Single, Double, DateTime, TimeSpan (Overrides |

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------|
| | | VelocityDbComparer(Key).SetComparisonArrayFromObject(Key,Byte[], Boolean).) |
|  | SetupFieldsToCompare | Initializes data used to compare fields as specified with this comparer. |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key) Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareByField(Key)(SessionBase) | This special constructor is provided so that VelocityDb can instantiate the object after reading it from disk. This is also the place to initialize transient variables of the object. |
|  | CompareByField(Key)(String, SessionBase, Boolean, Boolean, Boolean) | Compares objects by comparing one or more data fields of type Key |

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key) Constructor (SessionBase)

This special constructor is provided so that VelocityDb can instantiate the object after reading it from disk. This is also the place to initialize transient variables of the object.

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public CompareByField(  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    session As SessionBase  
)
```

C++

```
public:  
CompareByField(  
    SessionBase^ session  
)
```

F#

```
new :  
    session : SessionBase -> CompareByField
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object.

See Also

[CompareByField\(Key\)Class](#)

[CompareByField\(Key\)Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key) Constructor (String, SessionBase, Boolean, Boolean, Boolean)

Compares objects by comparing one or more data fields of type Key

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public CompareByField(  
    string field,  
    SessionBase session,  
    bool doIdCompareIfEqual = false,  
    bool sortStringsByHashCode = false,  
    bool ascendingSort = true  
)
```

VB

```
Public Sub New (  
    field As String,  
    session As SessionBase,  
    Optional doIdCompareIfEqual As Boolean = false,  
    Optional sortStringsByHashCode As Boolean = false,  
    Optional ascendingSort As Boolean = true  
)
```

C++

```
public:  
CompareByField(  
    String^ field,  
    SessionBase^ session,  
    bool doIdCompareIfEqual = false,  
    bool sortStringsByHashCode = false,  
    bool ascendingSort = true  
)
```

F#

```
new :  
    field : string *  
    session : SessionBase *  
    ?doIdCompareIfEqual : bool *  
    ?sortStringsByHashCode : bool *  
    ?ascendingSort : bool  
(* Defaults:  
    let _doIdCompareIfEqual = defaultArg doIdCompareIfEqual false  
    let _sortStringsByHashCode = defaultArg sortStringsByHashCode false  
    let _ascendingSort = defaultArg ascendingSort true  
)  
-> CompareByField
```

Parameters

field

Type: [System.String](#)

Primary field to use for comparison.

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

doldCompareIfEqual (Optional)

Type: [System.Boolean](#)

Optionally compare Oid's when fields are equal between compared objects

sortStringsByHashCode (Optional)

Type: [System.Boolean](#)

Sort string fields by hash code first then secondly as normal string compare

ascendingSort (Optional)

Type: [System.Boolean](#)

If `true` (`True` in Visual Basic) (default), sort in ascending order, otherwise in descending order

See Also

[CompareByField\(Key\)Class](#)




[CompareByField\(Key\)Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).CompareByField(Key) Properties

The [CompareByField\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddOidCompare | Is the object Oid going to be compared when all the compared fields are equal? (Overrides VelocityDbComparer(Key).AddOidCompare.) |
|  | FieldsToCompare | |
|  | OidPartOfComparisonBytes | Is the Oid (or OidShort) included in the comparison array bytes? (Overrides VelocityDbComparer(Key).OidPartOfComparisonBytes.) |

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).AddOidCompare Property

Is the object Oid going to be compared when all the compared fields are equal?

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AddOidCompare { get; }
```

VB

```
Public Overrides ReadOnly Property AddOidCompare As Boolean  
    Get
```

C++

```
public:  
virtual property bool AddOidCompare {  
    bool get () override;  
}
```

F#

```
abstract AddOidCompare : bool with get  
override AddOidCompare : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).FieldsToCompare Property

[Missing <summary> documentation for "P:VelocityDb.Collection.Comparer.CompareByField`1.FieldsToCompare"]

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DataMember[] FieldsToCompare { get; }
```

VB

```
Public ReadOnly Property FieldsToCompare As DataMember()  
    Get
```

C++

```
public:  
property array<DataMember^>^ FieldsToCompare {  
    array<DataMember^>^ get ();  
}
```

F#

```
member FieldsToCompare : DataMember[] with get
```

Property Value

Type: [DataMember\[\]](#)

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).OidPartOfComparisonBytes Property

Is the Oid (or OidShort) included in the comparison array bytes?

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool OidPartOfComparisonBytes { get; }
```

VB

```
Public Overrides ReadOnly Property OidPartOfComparisonBytes As Boolean  
    Get
```

C++

```
public:  
virtual property bool OidPartOfComparisonBytes {  
    bool get () override;  
}
```

F#

```
abstract OidPartOfComparisonBytes : bool with get  
override OidPartOfComparisonBytes : bool with get
```

Property Value

Type: [Boolean](#)

See Also










[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)



CompareByField(Key).CompareByField(Key) Methods

The [CompareByField\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddFieldToCompare | Add an additional field to compare when prior fields all are equal |
|  Compare(Key, Key) | Compares field(s) of Key a with field(s) of Key b (Overrides VelocityDbComparer(Key).Compare(Key, Key) .) |
|  Compare(Byte[], Int32,Byte[], Key) | (Overrides VelocityDbComparer(Key).Compare(Byte[], Int32,Byte[], Key) .) |
|  CompareField | For internal use. Compares a field value in two objects |
|  GetHashCode32  | Microsoft messed up the 64 bit version of GetHashCode. Use this original 32-bit version to get consistent String hash codes across 32 bit and 64 bit builds. See remarks in https://msdn.microsoft.com/en-us/library/system.string.gethashcode(v=vs.110).aspx |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase) .) |
|  SetComparisonArrayFromObject | Sets a byte array equivalent to the field values to compare. Using this bypasses object compares, instead the byte arrays are compared byte by byte until a diff is found. If a byte diff isn't found then object compares may happen. Field types currently supported: byte, bool, Char, string, Int16, Int32, Int64, UInt16, UInt32, UInt64, Single, Double, DateTime, TimeSpan (Overrides VelocityDbComparer(Key).SetComparisonArrayFromObject(Key,Byte[], Boolean) .) |
|  SetupFieldsToCompare | Initializes data used to compare fields as specified with this comparer. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).AddFieldToCompare Method

Add an additional field to compare when prior fields all are equal

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddFieldToCompare (  
    string field,  
    bool ascendingSort = true  
)
```

VB

```
Public Sub AddFieldToCompare (  
    field As String,  
    Optional ascendingSort As Boolean = true  
)
```

C++

```
public:  
void AddFieldToCompare (  
    String^ field,  
    bool ascendingSort = true  
)
```

F#

```
member AddFieldToCompare :  
    field : string *  
    ?ascendingSort : bool  
(* Defaults:  
    let_ascendingSort = defaultArg ascendingSort true  
)  
-> unit
```

Parameters

field

Type: [System.String](#)

Field name of field to compare

ascendingSort (Optional)

Type: [System.Boolean](#)



If `true` (`True` in Visual Basic) (default), sort in ascending order, otherwise in descending order

See Also

[CompareByField\(Key\)Class](#)

CompareByField(Key).Compare Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
|  | Compare(Key, Key) | Compares field(s) of Key a with field(s) of Key b (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |
|  | Compare(Byte[], Int32, Byte[], Key) | (Overrides VelocityDbComparer(Key).Compare(Byte[], Int32, Byte[], Key).) |

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).Compare Method (Key, Key)

Compares field(s) of Key a with field(s) of Key b

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Compare(  
    Key a,  
    Key b  
)
```

VB

```
Public Overrides Function Compare (  
    a As Key,  
    b As Key  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    Key a,  
    Key b  
) override
```

F#

```
abstract Compare :  
    a : 'Key *  
    b : 'Key -> int  
override Compare :  
    a : 'Key *  
    b : 'Key -> int
```

Parameters

a

Type: *Key*
the first object

b

Type: *Key*
the second object

Return Value

Type: [Int32](#)

-1 if a less than b, 0 if equal and 1 if a greater than b

VelocityDB Class Library

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[CompareByField\(Key\)Class](#)

[Compare Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).Compare Method (Byte[], Int32, Byte[], Key)

[Missing <summary> documentation for

"M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Compare(  
    byte[] nodeComparisonArray,  
    int nodeArraySkip,  
    byte[] comparisonArray,  
    Key key  
)
```

VB

```
Public Overrides Function Compare (  
    nodeComparisonArray As Byte(),  
    nodeArraySkip As Integer,  
    comparisonArray As Byte(),  
    key As Key  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    array<unsigned char>^ nodeComparisonArray,  
    int nodeArraySkip,  
    array<unsigned char>^ comparisonArray,  
    Key key  
) override
```

F#

```
abstract Compare :  
    nodeComparisonArray : byte[] *  
    nodeArraySkip : int *  
    comparisonArray : byte[] *  
    key : 'Key -> int  
override Compare :  
    nodeComparisonArray : byte[] *  
    nodeArraySkip : int *  
    comparisonArray : byte[] *  
    key : 'Key -> int
```

Parameters

nodeComparisonArray

Type: [System.Byte\[\]](#)

[Missing <param name="nodeComparisonArray"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

nodeArraySkip

Type: [System.Int32](#)

[Missing <param name="nodeArraySkip"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

comparisonArray

Type: [System.Byte\[\]](#)

[Missing <param name="comparisonArray"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

key

Type: *Key*

[Missing <param name="key"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDb.Collection.Comparer.CompareByField`1.Compare(System.Byte[],System.Int32,System.Byte[],`0)"]

See Also

[CompareByField\(Key\)Class](#)

[Compare Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).CompareField Method

For internal use. Compares a field value in two objects

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareField(  
    DataMember member,  
    Object a,  
    Object b,  
    int fieldIndex  
)
```

VB

```
Public Function CompareField (  
    member As DataMember,  
    a As Object,  
    b As Object,  
    fieldIndex As Integer  
) As Integer
```

C++

```
public:  
int CompareField(  
    DataMember^ member,  
    Object^ a,  
    Object^ b,  
    int fieldIndex  
)
```

F#

```
member CompareField :  
    member : DataMember *  
    a : Object *  
    b : Object *  
    fieldIndex : int -> int
```

Parameters

member

Type: [VelocityDb.TypeInfo.DataMember](#)

Field meta data

a

Type: [System.Object](#)

First object

b

Type: [System.Object](#)

Second object

fieldIndex

Type: [System.Int32](#)

indicates which of the fields in this comparer

Return Value

Type: [Int32](#)

A 32-bit signed integer that indicates whether this instance precedes, follows, or appears in the same position in the sort order as the value parameter.

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).GetHashCode32 Method

Microsoft messed up the 64 bit version of GetHashCode. Use this original 32-bit version to get consistent [String](#) hash codes across 32 bit and 64 bit builds. See remarks in [https://msdn.microsoft.com/en-us/library/system.string.gethashcode\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string.gethashcode(v=vs.110).aspx)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int GetHashCode32 (  
    string s  
)
```

VB

```
Public Shared Function GetHashCode32 (  
    s As String  
) As Integer
```

C++

```
public:  
static int GetHashCode32 (  
    String^ s  
)
```

F#

```
static member GetHashCode32 :  
    s : string -> int
```

Parameters

s

Type: [System.String](#)

[String](#)

to compute hash code for.

Return Value

Type: [Int32](#)

A 32-bit hash code

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).SetComparisonArrayFromObject Method

Sets a byte array equivalent to the field values to compare. Using this bypasses object compares, instead the byte arrays are compared byte by byte until a diff is found. If a byte diff isn't found then object compares may happen. Field types currently supported: byte, bool, Char, string, Int16, Int32, Int64, UInt16, UInt32, UInt64, Single, Double, DateTime, TimeSpan

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetComparisonArrayFromObject (
    Key key,
    byte[] comparisonArray,
    bool oidShort
)
```

VB

```
Public Overrides Sub SetComparisonArrayFromObject (
    key As Key,
    comparisonArray As Byte(),
    oidShort As Boolean
)
```

C++

```
public:
virtual void SetComparisonArrayFromObject (
    Key key,
    array<unsigned char>^ comparisonArray,
    bool oidShort
) override
```

F#

```
abstract SetComparisonArrayFromObject :
    key : 'Key *
    comparisonArray : byte[] *
    oidShort : bool -> unit
override SetComparisonArrayFromObject :
    key : 'Key *
    comparisonArray : byte[] *
    oidShort : bool -> unit
```

Parameters

key

Type: *Key*

A key from which to extract a comparison byte array based on field(s) to compare

comparisonArray

VelocityDB Class Library

Type: [System.Byte\[\]](#)

The array to set bytes of

oidShort

Type: [System.Boolean](#)

If *oidShort* is true then only the page and slot numbers need to be compared if the comparator bytes includes any Oid bytes

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByField(Key).SetupFieldsToCompare Method

Initializes data used to compare fields as specified with this comparer.

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void SetupFieldsToCompare ()
```

VB

```
Public Overridable Sub SetupFieldsToCompare
```

C++

```
public:  
virtual void SetupFieldsToCompare ()
```

F#

```
abstract SetupFieldsToCompare : unit -> unit  
override SetupFieldsToCompare : unit -> unit
```

See Also

[CompareByField\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByFieldIndex(Key) Class

Internal use for Index implementation

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.Comparer.VelocityDbComparer\(Key\)](#)

[VelocityDb.Collection.Comparer.CompareByField\(Key\)](#)

VelocityDb.Collection.Comparer.CompareByFieldIndex(Key)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class CompareByFieldIndex<Key> : CompareByField<Key>
```

VB

```
<SerializableAttribute>  
Public Class CompareByFieldIndex(Of Key)  
    Inherits CompareByField(Of Key)
```

C++

```
[SerializableAttribute]  
generic<typename Key>  
public ref class CompareByFieldIndex : public CompareByField<Key>
```

F#

```
[<SerializableAttribute>]  
type CompareByFieldIndex<'Key> =  
    class  
        inherit CompareByField<'Key>  
    end
```


Type Parameters

Key




[Missing <typeparam name="Key"/> documentation for "T:VelocityDb.Collection.Comparer.CompareByFieldIndex`1"]

The CompareByFieldIndex(Key) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|------------------------------------------------------------------|
|  | CompareByFieldIndex(Key) | Initializes a new instance of the CompareByFieldIndex(Key) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare | Compares field(s) of Key a with field(s) of Key b (Overrides CompareByField(Key).Compare(Key, Key).) |
|  | InitNew | Internal Use |
|  | SetupFieldsToCompare | Initializes data used to compare fields as specified with this comparer. (Overrides CompareByField(Key).SetupFieldsToCompare().) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.Comparer Namespace](#)

CompareByFieldIndex(Key) Constructor

Initializes a new instance of the [CompareByFieldIndex\(Key\)](#) class

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public CompareByFieldIndex(
    TypeVersion tVersion,
    string fields,
    SessionBase session,
    bool doIdCompareIfEqual = false,
    bool sortStringsByHashCode = false
)
```

VB

```
Public Sub New (
    tVersion As TypeVersion,
    fields As String,
    session As SessionBase,
    Optional doIdCompareIfEqual As Boolean = false,
    Optional sortStringsByHashCode As Boolean = false
)
```

C++

```
public:
CompareByFieldIndex(
    TypeVersion^ tVersion,
    String^ fields,
    SessionBase^ session,
    bool doIdCompareIfEqual = false,
    bool sortStringsByHashCode = false
)
```

F#

```
new :
    tVersion : TypeVersion *
    fields : string *
    session : SessionBase *
    ?doIdCompareIfEqual : bool *
    ?sortStringsByHashCode : bool
(* Defaults:
    let _doIdCompareIfEqual = defaultArg doIdCompareIfEqual false
    let _sortStringsByHashCode = defaultArg sortStringsByHashCode false
*)
-> CompareByFieldIndex
```

Parameters

tVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="tVersion"/> documentation for
"M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.#ctor(VelocityDb.TypeInfo.TypeVersion, System.String, VelocityDb.Session.SessionBase, System.Boolean, System.Boolean)"]

fields

Type: [System.String](#)

[Missing <param name="fields"/> documentation for
"M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.#ctor(VelocityDb.TypeInfo.TypeVersion, System.String, VelocityDb.Session.SessionBase, System.Boolean, System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for
"M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.#ctor(VelocityDb.TypeInfo.TypeVersion, System.String, VelocityDb.Session.SessionBase, System.Boolean, System.Boolean)"]

doldCompareIfEqual (Optional)

Type: [System.Boolean](#)

[Missing <param name="doldCompareIfEqual"/> documentation for
"M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.#ctor(VelocityDb.TypeInfo.TypeVersion, System.String, VelocityDb.Session.SessionBase, System.Boolean, System.Boolean)"]

sortStringsByHashCode (Optional)

Type: [System.Boolean](#)

[Missing <param name="sortStringsByHashCode"/> documentation for
"M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.#ctor(VelocityDb.TypeInfo.TypeVersion, System.String, VelocityDb.Session.SessionBase, System.Boolean, System.Boolean)"]

See Also




[CompareByFieldIndex\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)



CompareByFieldIndex(Key).CompareByFieldIndex(Key) Methods

The [CompareByFieldIndex\(Key\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare | Compares field(s) of Key a with field(s) of Key b (Overrides CompareByField(Key).Compare(Key, Key).) |
|  | InitNew | Internal Use |
|  | SetupFieldsToCompare | Initializes data used to compare fields as specified with this comparer. (Overrides CompareByField(Key).SetupFieldsToCompare().) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[CompareByFieldIndex\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByFieldIndex(Key).Compare Method

Compares field(s) of Key a with field(s) of Key b

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Compare(  
    Key a,  
    Key b  
)
```

VB

```
Public Overrides Function Compare (  
    a As Key,  
    b As Key  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    Key a,  
    Key b  
) override
```

F#

```
abstract Compare :  
    a : 'Key *  
    b : 'Key -> int  
override Compare :  
    a : 'Key *  
    b : 'Key -> int
```

Parameters

a

Type: *Key*
the first object

b

Type: *Key*
the second object

Return Value

Type: [Int32](#)

-1 if a less than b, 0 if equal and 1 if a greater than b

VelocityDB Class Library

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[CompareByFieldIndex\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByFieldIndex(Key).InitNew Method

Internal Use

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InitNew(  
    TypeVersion tVersion,  
    string fields  
)
```

VB

```
Public Sub InitNew (  
    tVersion As TypeVersion,  
    fields As String  
)
```

C++

```
public:  
void InitNew(  
    TypeVersion^ tVersion,  
    String^ fields  
)
```

F#

```
member InitNew :  
    tVersion : TypeVersion *  
    fields : string -> unit
```

Parameters

tVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="tVersion"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.InitNew(VelocityDb.TypeInfo.TypeVersion,System.String)"]

fields

Type: [System.String](#)

[Missing <param name="fields"/> documentation for "M:VelocityDb.Collection.Comparer.CompareByFieldIndex`1.InitNew(VelocityDb.TypeInfo.TypeVersion,System.String)"]

VelocityDB Class Library

See Also

[CompareByFieldIndex\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

CompareByFieldIndex(Key).SetupFieldsToCompare Method

Initializes data used to compare fields as specified with this comparer.

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetupFieldsToCompare ()
```

VB

```
Public Overrides Sub SetupFieldsToCompare
```

C++

```
public:  
virtual void SetupFieldsToCompare () override
```

F#

```
abstract SetupFieldsToCompare : unit -> unit  
override SetupFieldsToCompare : unit -> unit
```

See Also

[CompareByFieldIndex\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

HashCodeComparer(T) Class

When the type of object being compared is a simple type like a string and you don't care about the sort order, then this comparator is what you want to use. It is similar to using a hash based collection but with less overhead (more compact) This kind of comparator especially shines when you create the BTreeSet/BTreeMap using a comparisonByteArray size of 4

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.Comparer.VelocityDbComparer\(T\)](#)

VelocityDb.Collection.Comparer.HashCodeComparer(T)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class HashCodeComparer<T> : VelocityDbComparer<T>  
where T : IComparable
```

VB

```
<SerializableAttribute>  
Public Class HashCodeComparer (Of T As IComparable)  
    Inherits VelocityDbComparer (Of T)
```

C++

```
[SerializableAttribute]  
generic<typename T>  
where T : IComparable  
public ref class HashCodeComparer : public VelocityDbComparer<T>
```

F#

```
[<SerializableAttribute>]  
type HashCodeComparer<'T when 'T : IComparable> =  
    class  
        inherit VelocityDbComparer<'T>  
    end
```


Type Parameters

T



The type of object to compare which must implement [IComparable](#)

The HashCodeComparer(T) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------------------|
|  | HashCodeComparer(T) | Initializes a new instance of the HashCodeComparer(T) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |
|  | SetComparisonArrayFromObject | Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes (Overrides VelocityDbComparer(Key).SetComparisonArrayFromObject(Key, Byte[], Boolean).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.Comparer Namespace](#)

HashCodeComparer(T) Constructor

Initializes a new instance of the [HashCodeComparer\(T\)](#) class

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public HashCodeComparer ()
```

VB

```
Public Sub New
```

C++

```
public:  
HashCodeComparer ()
```

F#

```
new : unit -> HashCodeComparer
```

See Also



[HashCodeComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)



HashCodeComparer(T).HashCodeComparer(T) Methods

The [HashCodeComparer\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Compare | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |
|  SetComparisonArrayFromObject | Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes (Overrides VelocityDbComparer(Key).SetComparisonArrayFromObject(Key, Byte[], Boolean).) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[HashCodeComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

HashCodeComparer(*T*).Compare Method

If the Key objects implements [IComparable](#) then the NodeKeys are compared that way, otherwise the [Id](#) are compared

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Compare(  
    T a,  
    T b  
)
```

VB

```
Public Overrides Function Compare (  
    a As T,  
    b As T  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    T a,  
    T b  
) override
```

F#

```
abstract Compare :  
    a : 'T *  
    b : 'T -> int  
override Compare :  
    a : 'T *  
    b : 'T -> int
```

Parameters

a

Type: *T*

object a to compare

b

Type: *T*

object b to compare with

Return Value

Type: [Int32](#)

-1 if a less than b, 0 if == and 1 if a is greater than b

VelocityDB Class Library

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[HashCodeComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

HashCodeComparer(T).SetComparisonArrayFromObject Method

Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetComparisonArrayFromObject (
    T key,
    byte[] comparisonArray,
    bool oidShort
)
```

VB

```
Public Overrides Sub SetComparisonArrayFromObject (
    key As T,
    comparisonArray As Byte(),
    oidShort As Boolean
)
```

C++

```
public:
virtual void SetComparisonArrayFromObject (
    T key,
    array<unsigned char>^ comparisonArray,
    bool oidShort
) override
```

F#

```
abstract SetComparisonArrayFromObject :
    key : 'T *
    comparisonArray : byte[] *
    oidShort : bool -> unit
override SetComparisonArrayFromObject :
    key : 'T *
    comparisonArray : byte[] *
    oidShort : bool -> unit
```

Parameters

key

Type: *T*

The object from which to extract the comparison array bytes

comparisonArray

Type: [System.Byte\[\]](#)

The array containing the object comparison bytes

VelocityDB Class Library

oidShort

Type: [System.Boolean](#)

If `oidShort` is true then only the page and slot numbers need to be compared if the comparator bytes includes any Oid bytes

See Also

[HashCodeComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

SimpleComparer(T) Class

When the type of object being compared is a simple type like string, int, ..., this is the comparator you probably want to use.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.Collection.Comparer.VelocityDbComparer\(T\)](#)

VelocityDb.Collection.Comparer.SimpleComparer(T)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SimpleComparer<T> : VelocityDbComparer<T>  
where T : IComparable
```

VB

```
<SerializableAttribute>  
Public Class SimpleComparer(Of T As IComparable)  
    Inherits VelocityDbComparer(Of T)
```

C++

```
[SerializableAttribute]  
generic<typename T>  
where T : IComparable  
public ref class SimpleComparer : public VelocityDbComparer<T>
```

F#

```
[<SerializableAttribute>]  
type SimpleComparer<'T when 'T : IComparable> =  
    class  
        inherit VelocityDbComparer<'T>  
    end
```


Type Parameters

T


The type of object to compare which must implement [IComparable](#)

The SimpleComparer(T) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------|
|  | SimpleComparer(T) | Initializes a new instance of the SimpleComparer(T) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.Comparer Namespace](#)

SimpleComparer(T) Constructor

Initializes a new instance of the [SimpleComparer\(T\)](#) class

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SimpleComparer ()
```

VB

```
Public Sub New
```

C++

```
public:  
SimpleComparer ()
```

F#

```
new : unit -> SimpleComparer
```

See Also


[SimpleComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)



SimpleComparer(T).SimpleComparer(T) Methods

The [SimpleComparer\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared (Overrides VelocityDbComparer(Key).Compare(Key, Key).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[SimpleComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

SimpleComparer(T).Compare Method

If the Key objects implements [IComparable](#) then the NodeKeys are compared that way, otherwise the [Id](#) are compared

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int Compare(  
    T a,  
    T b  
)
```

VB

```
Public Overrides Function Compare (  
    a As T,  
    b As T  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    T a,  
    T b  
) override
```

F#

```
abstract Compare :  
    a : 'T *  
    b : 'T -> int  
override Compare :  
    a : 'T *  
    b : 'T -> int
```

Parameters

a

Type: *T*

object a to compare

b

Type: *T*

object b to compare with

Return Value

Type: [Int32](#)

-1 if a less than b, 0 if == and 1 if a is greater than b

VelocityDB Class Library

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[SimpleComparer\(T\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key) Class

The base class of for comparing persistent objects. If the Key objects implements [IComparable](#) then the NodeKeys are compared that way, otherwise the [Id](#) are compared

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Collection.Comparer.VelocityDbComparer(Key)

[VelocityDb.Collection.Comparer.CompareByField\(Key\)](#)

[VelocityDb.Collection.Comparer.HashCodeComparer\(T\)](#)

[VelocityDb.Collection.Comparer.SimpleComparer\(T\)](#)

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VelocityDbComparer<Key> : OptimizedPersistable,  
    IComparer<Key>
```

VB

```
<SerializableAttribute>  
Public Class VelocityDbComparer (Of Key)  
    Inherits OptimizedPersistable  
    Implements IComparer (Of Key)
```

C++

```
[SerializableAttribute]  
generic<typename Key>  
public ref class VelocityDbComparer : public OptimizedPersistable,  
    IComparer<Key>
```

F#

```
[<SerializableAttribute>]  
type VelocityDbComparer<'Key> =  
    class  
        inherit OptimizedPersistable  
        interface IComparer<'Key>  
    end
```


Type Parameters

Key



The type of keys to compare

The VelocityDbComparer(Key) type exposes the following members.




Constructors

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
|  VelocityDbComparer(Key) | Initializes a new instance of the VelocityDbComparer(Key) class |



Properties

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  AddOidCompare | The default compare compares the object Oid's but subclasses may not want to do so as a last compare before declaring objects as being equal. |
|  OidPartOfComparisonBytes | Is the Oid part of the comparison bytes? If so, we don't need to do a separate Oid compare when Oid is part of compare |

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|  Compare(Key, Key) | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared |
|  Compare(Byte[], Int32,Byte[], Key) | Internal use, compares comparison array bytes |
|  SetComparisonArrayFromObject | Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key) Constructor

Initializes a new instance of the [VelocityDbComparer\(Key\)](#) class

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbComparer ()
```

VB

```
Public Sub New
```

C++

```
public:  
VelocityDbComparer ()
```

F#

```
new : unit -> VelocityDbComparer
```

See Also



[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).VelocityDbComparer(Key) Properties

The [VelocityDbComparer\(Key\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddOidCompare | The default compare compares the object Oid's but subclasses may not want to do so as a last compare before declaring objects as being equal. |
|  | OidPartOfComparisonBytes | Is the Oid part of the comparison bytes? If so, we don't need to do a separate Oid compare when Oid is part of compare |

See Also

[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).AddOidCompare Property

The default compare compares the object Oid's but subclasses may not want to do so as a last compare before declaring objects as being equal.

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool AddOidCompare { get; }
```

VB

```
Public Overridable ReadOnly Property AddOidCompare As Boolean  
    Get
```

C++

```
public:  
virtual property bool AddOidCompare {  
    bool get ();  
}
```

F#

```
abstract AddOidCompare : bool with get  
override AddOidCompare : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).OidPartOfComparisonBytes Property

Is the Oid part of the comparison bytes? If so, we don't need to do a separate Oid compare when Oid is part of compare

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool OidPartOfComparisonBytes { get; }
```

VB

```
Public Overridable ReadOnly Property OidPartOfComparisonBytes As Boolean  
    Get
```

C++

```
public:  
virtual property bool OidPartOfComparisonBytes {  
    bool get ();  
}
```

F#

```
abstract OidPartOfComparisonBytes : bool with get  
override OidPartOfComparisonBytes : bool with get
```

Property Value

Type: [Boolean](#)

See Also




[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)



VelocityDbComparer(Key).VelocityDbComparer(Key) Methods

The [VelocityDbComparer\(Key\)](#) generic type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|  Compare(Key, Key) | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared |
|  Compare(Byte[], Int32, Byte[], Key) | Internal use, compares comparison array bytes |
|  SetComparisonArrayFromObject | Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |



See Also

[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).Compare Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compare(Key, Key) | If the Key objects implements IComparable then the NodeKeys are compared that way, otherwise the Id are compared |
|  | Compare(Byte[], Int32, Byte[], Key) | Internal use, compares comparison array bytes |

See Also

[VelocityDbComparer\(Key\)Class](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).Compare Method (Key, Key)

If the Key objects implements [IComparable](#) then the NodeKeys are compared that way, otherwise the [Id](#) are compared

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual int Compare(  
    Key a,  
    Key b  
)
```

VB

```
Public Overridable Function Compare (  
    a As Key,  
    b As Key  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    Key a,  
    Key b  
)
```

F#

```
abstract Compare :  
    a : 'Key *  
    b : 'Key -> int  
override Compare :  
    a : 'Key *  
    b : 'Key -> int
```

Parameters

a

Type: *Key*

object a to compare

b

Type: *Key*

object b to compare with

Return Value

Type: [Int32](#)

-1 if a less than b, 0 if == and 1 if a is greater than b

VelocityDB Class Library

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[VelocityDbComparer\(Key\)Class](#)

[Compare Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).Compare Method (Byte[], Int32, Byte[], Key)

Internal use, compares comparison array bytes

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual int Compare(  
    byte[] nodeComparisonArray,  
    int index,  
    byte[] comparisonArray,  
    Key key  
)
```

VB

```
Public Overridable Function Compare (  
    nodeComparisonArray As Byte(),  
    index As Integer,  
    comparisonArray As Byte(),  
    key As Key  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    array<unsigned char>^ nodeComparisonArray,  
    int index,  
    array<unsigned char>^ comparisonArray,  
    Key key  
)
```

F#

```
abstract Compare :  
    nodeComparisonArray : byte[] *  
    index : int *  
    comparisonArray : byte[] *  
    key : 'Key -> int  
override Compare :  
    nodeComparisonArray : byte[] *  
    index : int *  
    comparisonArray : byte[] *  
    key : 'Key -> int
```

Parameters

nodeComparisonArray

Type: [System.Byte\[\]](#)

BTree node array of [Byte](#) for all current objects in node

VelocityDB Class Library

index

Type: [System.Int32](#)

Object index in node to compare with

comparisonArray

Type: [System.Byte\[\]](#)

Bytes to compare with

key

Type: *Key*

Object compared with

Return Value

Type: [Int32](#)

-1 if this node bytes is less than comparisonArray bytes, 0 if equal, otherwise 1

See Also

[VelocityDbComparer\(Key\)Class](#)

[Compare Overload](#)

[VelocityDb.Collection.Comparer Namespace](#)

VelocityDbComparer(Key).SetComparisonArrayFromObject Method

Comparators that use comparison arrays need to override this function. Given a Key object, set the comparisonArray bytes

Namespace: [VelocityDb.Collection.Comparer](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void SetComparisonArrayFromObject (
    Key key,
    byte[] comparisonArray,
    bool oidShort
)
```

VB

```
Public Overridable Sub SetComparisonArrayFromObject (
    key As Key,
    comparisonArray As Byte(),
    oidShort As Boolean
)
```

C++

```
public:
virtual void SetComparisonArrayFromObject (
    Key key,
    array<unsigned char>^ comparisonArray,
    bool oidShort
)
```

F#

```
abstract SetComparisonArrayFromObject :
    key : 'Key *
    comparisonArray : byte[] *
    oidShort : bool -> unit
override SetComparisonArrayFromObject :
    key : 'Key *
    comparisonArray : byte[] *
    oidShort : bool -> unit
```

Parameters

key

Type: *Key*

The object from which to extract the comparison array bytes

comparisonArray

Type: [System.Byte\[\]](#)

The array containing the object comparison bytes

VelocityDB Class Library

oidShort

Type: [System.Boolean](#)

If *oidShort* is true then only the page and slot numbers need to be compared if the comparator bytes includes any Oid bytes

See Also










[VelocityDbComparer\(Key\)Class](#)








[VelocityDb.Collection.Comparer Namespace](#)












VelocityDb.Exceptions Namespace












The `VelocityDb.Session` namespace contains all exceptions that may be thrown by VelocityDB code.

Classes

| Class | Description |
|------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|  AlreadyInCommitException | Exception thrown when attempting to commit a transaction when we already are within transaction commit code. |
|  AlreadyInTransactionException | Exception thrown when attempting to start a transaction when an active transaction already exist. |
|  DatabaseAlreadyExistsException | Exception thrown when trying to create a Database that already exist (file system file exist). |
|  DatabaseDoesNotExistException | Exception thrown when the requested Database does not exist. |
|  DatabaseReadLockException | Exception thrown when a Database level read lock failed. |
|  DesKeyMissingException | Exception thrown when an internal error was detected. This type of error is most likly due to a bug in VelocityDb. |
|  FieldDoesNotExistException | Exception thrown when specifying a field name that does not exist in the given class/struct. |
|  IndexDatabaseNotSpecifiedException | Exception thrown when attempting to open an in database index without specifying a database. |
|  IndexDatabaseOrBTreeMissingException | Exception thrown when attempting to open an in database index without specifying a database. |

| | | |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | IndexDatabaseSpecifiedForGlobalIndexException | <p>Exception thrown when attempting to open a global index and specifying a database.</p> |
|  | InternalErrorException | <p>Exception thrown when an internal error was detected. This type of error is most likely due to a bug in VelocityDb.</p> |
|  | InTransactionException | <p>Exception thrown when calling certain SessionBase api that requires that session IS NOT in an active transaction.</p> |
|  | InUpdateTransactionException | <p>Exception thrown when calling certain SessionBase api that requires that session IS NOT in an active update transaction.</p> |
|  | InvalidChangeOfDatabaseLocation | <p>Exception thrown when trying to do an invalid change of a DatabaseLocation. It could be a change to the start database number that if permitted skips one or more other DatabaseLocations.</p> |
|  | InvalidChangeOfDefaultLocationException | <p>Exception thrown when trying to do an invalid change of the startup DatabaseLocation.</p> |
|  | MaxNumberOfDatabasesException | <p>Exception thrown when trying to create a Database but no more Database numbers are available. The Maximum Database number is MaxValue or 13 if no valid license is found.</p> |
|  | NotInTransactionException | <p>Exception thrown when trying to do persistent operations without first starting a transaction.</p> |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|  NoValidVelocityDBLicenseFoundException | Exception thrown when trying to add/update database schema and no valid VelocityDB license is found in license database (4.odt). |
|  NullObjectException | Exception thrown when an object unexpectedly is null. |
|  ObjectDoesNotExistException | Exception thrown when the requested >IOptimizedPersistable does not exist. |
|  ObjectNotInSameDatabaseAsOidShortCollectionException | Exception thrown when object is not in the same database as a ShortOid collection Database failed. |
|  OpenDatabaseException | Exception thrown when opening a Database failed. |
|  OptimisticLockingFailed | Exception thrown when a optimistic locking failed due to another transaction having updated a Page or Database. |
|  PageDeadLockException | Exception thrown when a Page level lock failed due to a deadlock with another transaction |
|  PageDoesNotExistException | Exception thrown when the requested >Page does not exist. |
|  PageReadLockException | Exception thrown when failing to get a Page level read lock. |
|  PageUpdateLockException | Exception thrown when a Page level update lock failed. |
|  PersistedObjectExpectedException | Exception thrown when a persisted object was expected but was not. |

| | | |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | ReferentialIntegrityException | Exception thrown when an internal error was detected. This type of error is most likely due to a bug in VelocityDb. |
|  | RequestedPlacementDatabaseNumberNotValidException | Exception thrown when a requested placement Database number is invalid. |
|  | RequestedPlacementPageNumberNotValidException | Exception thrown when a requested placement Page number is invalid. |
|  | SubscriptionsNotAvailableWithNoServerSessionException | Exception thrown when trying to use event subscription api with NON server session. Subscriptions require use of server. |
|  | SystemDatabaseNotFoundWithReadOnlyTransactionException | Exception thrown when a read only transaction try to open a non existing Database 0 |
|  | TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException | Exception thrown when attempting to start a read only transaction while in an update transaction. |
|  | TryingToDeleteDeletedDatabaseException | Exception thrown when attempting to delete a Database that already was deleted in the same transaction |
|  | UnexpectedException | Exception thrown when something unexpected happened and no specific exception exist for it. |
|  | UniqueConstraintException | Exception thrown when attempting to open an in database index without specifying a database. |
|  | UpdateLockFailedException | Exception thrown when an update lock failed. |
|  | WeakIOptimizedPersistableReferenceMustBePersistentException | Exception thrown when a persisted object was expected but was not. |

AlreadyInCommitException Class

Exception thrown when attempting to commit a transaction when we already are within transaction commit code.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.AlreadyInCommitException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class AlreadyInCommitException : Exception
```

VB

```
<SerializableAttribute>  
Public Class AlreadyInCommitException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class AlreadyInCommitException : public Exception
```

F#

```
[<SerializableAttribute>]  
type AlreadyInCommitException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

AlreadyInTransactionException Class

Exception thrown when attempting to start a transaction when an active transaction already exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.AlreadyInTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class AlreadyInTransactionException : Exception
```

VB

```
<SerializableAttribute>  
Public Class AlreadyInTransactionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class AlreadyInTransactionException : public Exception
```

F#

```
[<SerializableAttribute>]  
type AlreadyInTransactionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Class

Exception thrown when trying to create a [Database](#) that already exist (file system file exist).

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.DatabaseAlreadyExistsException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax





```
C#
[SerializableAttribute]
public class DatabaseAlreadyExistsException : Exception
```

```
VB
<SerializableAttribute>
Public Class DatabaseAlreadyExistsException
    Inherits Exception
```

```
C++
[SerializableAttribute]
public ref class DatabaseAlreadyExistsException : public Exception
```

```
F#
[<SerializableAttribute>]
type DatabaseAlreadyExistsException =
    class
        inherit Exception
    end
```

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
|  | DatabaseAlreadyExistsException() | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(String) | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(SerializationInfo, StreamingContext) | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(String, Exception) | Initializes a new instance of the DatabaseAlreadyExistsException class |





VelocityDB Class Library

See Also

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
|  | DatabaseAlreadyExistsException() | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(String) | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(SerializationInfo, StreamingContext) | Initializes a new instance of the DatabaseAlreadyExistsException class |
|  | DatabaseAlreadyExistsException(String, Exception) | Initializes a new instance of the DatabaseAlreadyExistsException class |

See Also

[DatabaseAlreadyExistsException Class](#)

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Constructor

Initializes a new instance of the [DatabaseAlreadyExistsException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseAlreadyExistsException ()
```

VB

```
Public Sub New
```

C++

```
public:  
DatabaseAlreadyExistsException ()
```

F#

```
new : unit -> DatabaseAlreadyExistsException
```

See Also

[DatabaseAlreadyExistsException Class](#)

[DatabaseAlreadyExistsException Overload](#)

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Constructor (String)

Initializes a new instance of the [DatabaseAlreadyExistsException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseAlreadyExistsException(  
    string message  
)
```

VB

```
Public Sub New (  
    message As String  
)
```

C++

```
public:  
DatabaseAlreadyExistsException(  
    String^ message  
)
```

F#

```
new :  
    message : string -> DatabaseAlreadyExistsException
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for "M:VelocityDb.Exceptions.DatabaseAlreadyExistsException.#ctor(System.String)"]

See Also

[DatabaseAlreadyExistsException Class](#)

[DatabaseAlreadyExistsException Overload](#)

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Constructor (SerializationInfo, StreamingContext)

Initializes a new instance of the [DatabaseAlreadyExistsException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseAlreadyExistsException(  
    SerializationInfo info,  
    StreamingContext context  
)
```

VB

```
Public Sub New (  
    info As SerializationInfo,  
    context As StreamingContext  
)
```

C++

```
public:  
DatabaseAlreadyExistsException(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

F#

```
new :  
    info : SerializationInfo *  
    context : StreamingContext -> DatabaseAlreadyExistsException
```

Parameters

info

Type: [System.Runtime.Serialization.SerializationInfo](#)

[Missing <param name="info"/> documentation for

"M:VelocityDb.Exceptions.DatabaseAlreadyExistsException.#ctor(System.Runtime.Serialization.SerializationInfo,System.Runtime.Serialization.StreamingContext)"]

context

Type: [System.Runtime.Serialization.StreamingContext](#)

[Missing <param name="context"/> documentation for

"M:VelocityDb.Exceptions.DatabaseAlreadyExistsException.#ctor(System.Runtime.Serialization.SerializationInfo,System.Runtime.Serialization.StreamingContext)"]

VelocityDB Class Library

See Also

[DatabaseAlreadyExistsException Class](#)

[DatabaseAlreadyExistsException Overload](#)

[VelocityDb.Exceptions Namespace](#)

DatabaseAlreadyExistsException Constructor (String, Exception)

Initializes a new instance of the [DatabaseAlreadyExistsException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseAlreadyExistsException(  
    string message,  
    Exception inner  
)
```

VB

```
Public Sub New (  
    message As String,  
    inner As Exception  
)
```

C++

```
public:  
DatabaseAlreadyExistsException(  
    String^ message,  
    Exception^ inner  
)
```

F#

```
new :  
    message : string *  
    inner : Exception -> DatabaseAlreadyExistsException
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for

"M:VelocityDb.Exceptions.DatabaseAlreadyExistsException.#ctor(System.String,System.Exception)"]

inner

Type: [System.Exception](#)

[Missing <param name="inner"/> documentation for

"M:VelocityDb.Exceptions.DatabaseAlreadyExistsException.#ctor(System.String,System.Exception)"]

See Also

[DatabaseAlreadyExistsException Class](#)

[DatabaseAlreadyExistsException Overload](#)

[VelocityDb.Exceptions Namespace](#)

DatabaseDoesNotExistException Class

Exception thrown when the requested >[Database](#) does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.DatabaseDoesNotExistException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class DatabaseDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class DatabaseDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class DatabaseDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type DatabaseDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

DatabaseReadLockException Class

Exception thrown when a [Database](#) level read lock failed.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.DatabaseReadLockException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class DatabaseReadLockException : Exception
```

VB

```
<SerializableAttribute>  
Public Class DatabaseReadLockException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class DatabaseReadLockException : public Exception
```

F#

```
[<SerializableAttribute>]  
type DatabaseReadLockException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

DesKeyMissingException Class

Exception thrown when an internal error was detected. This type of error is most likely due to a bug in VelocityDb.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.DesKeyMissingException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class DesKeyMissingException : Exception
```

VB

```
<SerializableAttribute>  
Public Class DesKeyMissingException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class DesKeyMissingException : public Exception
```

F#

```
[<SerializableAttribute>]  
type DesKeyMissingException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

FieldDoesNotExistException Class

Exception thrown when specifying a field name that does not exist in the given class/struct.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.FieldDoesNotExistException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class FieldDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class FieldDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class FieldDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type FieldDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

IndexDatabaseNotSpecifiedException Class

Exception thrown when attempting to open an in database index without specifying a database.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.IndexDatabaseNotSpecifiedException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class IndexDatabaseNotSpecifiedException : Exception
```

VB

```
<SerializableAttribute>  
Public Class IndexDatabaseNotSpecifiedException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class IndexDatabaseNotSpecifiedException : public Exception
```

F#

```
[<SerializableAttribute>]  
type IndexDatabaseNotSpecifiedException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

IndexDatabaseOrBTreeMissingException Class

Exception thrown when attempting to open an in database index without specifying a database.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.IndexDatabaseOrBTreeMissingException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class IndexDatabaseOrBTreeMissingException : Exception
```

VB

```
<SerializableAttribute>  
Public Class IndexDatabaseOrBTreeMissingException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class IndexDatabaseOrBTreeMissingException : public Exception
```

F#

```
[<SerializableAttribute>]  
type IndexDatabaseOrBTreeMissingException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

IndexDatabaseSpecifiedForGlobalIndexException Class

Exception thrown when attempting to open a global index and specifying a database.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.IndexDatabaseSpecifiedForGlobalIndexException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class IndexDatabaseSpecifiedForGlobalIndexException : Exception
```

VB

```
<SerializableAttribute>  
Public Class IndexDatabaseSpecifiedForGlobalIndexException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class IndexDatabaseSpecifiedForGlobalIndexException : public  
Exception
```

F#

```
[<SerializableAttribute>]  
type IndexDatabaseSpecifiedForGlobalIndexException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

InternalErrorException Class

Exception thrown when an internal error was detected. This type of error is most likely due to a bug in VelocityDb.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.InternalErrorException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InternalErrorException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InternalErrorException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InternalErrorException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InternalErrorException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

InTransactionException Class

Exception thrown when calling certain SessionBase api that requires that session IS NOT in an active transaction.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.InTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InTransactionException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InTransactionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InTransactionException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InTransactionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

InUpdateTransactionException Class

Exception thrown when calling certain SessionBase api that requires that session IS NOT in an active update transaction.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.InUpdateTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InUpdateTransactionException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InUpdateTransactionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InUpdateTransactionException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InUpdateTransactionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

InvalidChangeOfDatabaseLocation Class

Exception thrown when trying to do an invalid change of a [DatabaseLocation](#). It could be a change to the start database number that if permitted skips one or more other DatabaseLocations.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.InvalidChangeOfDatabaseLocation

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidChangeOfDatabaseLocation : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidChangeOfDatabaseLocation  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidChangeOfDatabaseLocation : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidChangeOfDatabaseLocation =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

InvalidChangeOfDefaultLocationException Class

Exception thrown when trying to do an invalid change of the startup [DatabaseLocation](#).

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.InvalidChangeOfDefaultLocationException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidChangeOfDefaultLocationException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidChangeOfDefaultLocationException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidChangeOfDefaultLocationException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidChangeOfDefaultLocationException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

MaxNumberOfDatabasesException Class

Exception thrown when trying to create a [Database](#) but no more Database numbers are available. The Maximum Database number is [MaxValue](#) or 13 if no valid license is found.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.MaxNumberOfDatabasesException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class MaxNumberOfDatabasesException : Exception
```

VB

```
<SerializableAttribute>  
Public Class MaxNumberOfDatabasesException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class MaxNumberOfDatabasesException : public Exception
```

F#

```
[<SerializableAttribute>]  
type MaxNumberOfDatabasesException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

NotInTransactionException Class

Exception thrown when trying to do persistent operations without first starting a transaction.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.NotInTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class NotInTransactionException : Exception
```

VB

```
<SerializableAttribute>  
Public Class NotInTransactionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class NotInTransactionException : public Exception
```

F#

```
[<SerializableAttribute>]  
type NotInTransactionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

NoValidVelocityDBLicenseFoundException Class

Exception thrown when trying to add/update database schema and no valid VelocityDB license is found in license database (4.odt).

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.NoValidVelocityDBLicenseFoundException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class NoValidVelocityDBLicenseFoundException : Exception
```

VB

```
<SerializableAttribute>  
Public Class NoValidVelocityDBLicenseFoundException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class NoValidVelocityDBLicenseFoundException : public Exception
```

F#

```
[<SerializableAttribute>]  
type NoValidVelocityDBLicenseFoundException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

NullObjectException Class

Exception thrown when an object unexpectedly is null.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.NullObjectException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class NullObjectException : Exception
```

VB

```
<SerializableAttribute>  
Public Class NullObjectException  
    Inherits Exception
```



C++

```
[SerializableAttribute]  
public ref class NullObjectException : public Exception
```

F#

```
[<SerializableAttribute>]  
type NullObjectException =  
    class  
        inherit Exception  
    end
```

Constructors



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------|
|  | NullObjectException() | Initializes a new instance of the NullObjectException class |
|  | NullObjectException(String) | Initializes a new instance of the NullObjectException class |

See Also

[VelocityDb.Exceptions Namespace](#)

NullObjectException Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------|
|  | NullObjectException() | Initializes a new instance of the NullObjectException class |
|  | NullObjectException(String) | Initializes a new instance of the NullObjectException class |

See Also

[NullObjectException Class](#)

[VelocityDb.Exceptions Namespace](#)

NullObjectException Constructor

Initializes a new instance of the [NullObjectException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public NullObjectException ()
```

VB

```
Public Sub New
```

C++

```
public:  
NullObjectException ()
```

F#

```
new : unit -> NullObjectException
```

See Also

[NullObjectException Class](#)

[NullObjectException Overload](#)

[VelocityDb.Exceptions Namespace](#)

NullObjectException Constructor (String)

Initializes a new instance of the [NullObjectException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public NullObjectException(  
    string message  
)
```

VB

```
Public Sub New (  
    message As String  
)
```

C++

```
public:  
NullObjectException(  
    String^ message  
)
```

F#

```
new :  
    message : string -> NullObjectException
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for "M:VelocityDb.Exceptions.NullObjectException.#ctor(System.String)"]

See Also

[NullObjectException Class](#)

[NullObjectException Overload](#)

[VelocityDb.Exceptions Namespace](#)

ObjectDoesNotExistException Class

Exception thrown when the requested >[IOptimizedPersistable](#) does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.ObjectDoesNotExistException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class ObjectDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class ObjectDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class ObjectDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type ObjectDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

ObjectNotInSameDatabaseAsOidShortCollectionException Class

Exception thrown when object is not in the same database as a ShortOid collection [Database](#) failed.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.ObjectNotInSameDatabaseAsOidShortCollectionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class ObjectNotInSameDatabaseAsOidShortCollectionException : Exception
```

VB

```
<SerializableAttribute>  
Public Class ObjectNotInSameDatabaseAsOidShortCollectionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class ObjectNotInSameDatabaseAsOidShortCollectionException :  
public Exception
```

F#

```
[<SerializableAttribute>]  
type ObjectNotInSameDatabaseAsOidShortCollectionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

OpenDatabaseException Class

Exception thrown when opening a [Database](#) failed.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.OpenDatabaseException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class OpenDatabaseException : Exception
```

VB

```
<SerializableAttribute>  
Public Class OpenDatabaseException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class OpenDatabaseException : public Exception
```

F#

```
[<SerializableAttribute>]  
type OpenDatabaseException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

OptimisticLockingFailed Class

Exception thrown when an optimistic locking failed due to another transaction having updated a Page or Database.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.OptimisticLockingFailed

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class OptimisticLockingFailed : Exception
```

VB

```
<SerializableAttribute>  
Public Class OptimisticLockingFailed  
    Inherits Exception
```

C++


```
[SerializableAttribute]  
public ref class OptimisticLockingFailed : public Exception
```

F#

```
[<SerializableAttribute>]  
type OptimisticLockingFailed =  
    class  
        inherit Exception  
    end
```

The **OptimisticLockingFailed** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------------------------|
|  | OptimisticLockingFailed | Initializes a new instance of the OptimisticLockingFailed class |

See Also

[VelocityDb.Exceptions Namespace](#)

OptimisticLockingFailed Constructor

Initializes a new instance of the [OptimisticLockingFailed](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public OptimisticLockingFailed(  
    string message  
)
```

```
VB  
Public Sub New (  
    message As String  
)
```

```
C++  
public:  
OptimisticLockingFailed(  
    String^ message  
)
```

```
F#  
new :  
    message : string -> OptimisticLockingFailed
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for "M:VelocityDb.Exceptions.OptimisticLockingFailed.#ctor(System.String)"]

See Also

[OptimisticLockingFailed Class](#)

[VelocityDb.Exceptions Namespace](#)

PageDeadLockException Class

Exception thrown when a [Page](#) level lock failed due to a deadlock with another transaction

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.PageDeadLockException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PageDeadLockException : Exception
```

VB

```
<SerializableAttribute>  
Public Class PageDeadLockException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class PageDeadLockException : public Exception
```

F#

```
[<SerializableAttribute>]  
type PageDeadLockException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

PageDoesNotExistException Class

Exception thrown when the requested >[Page](#) does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.PageDoesNotExistException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PageDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class PageDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class PageDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type PageDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

PageReadLockException Class

Exception thrown when failing to get a [Page](#) level read lock.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.PageReadLockException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PageReadLockException : Exception
```

VB

```
<SerializableAttribute>  
Public Class PageReadLockException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class PageReadLockException : public Exception
```

F#

```
[<SerializableAttribute>]  
type PageReadLockException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

PageUpdateLockException Class

Exception thrown when a [Page](#) level update lock failed.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.PageUpdateLockException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PageUpdateLockException : Exception
```

VB

```
<SerializableAttribute>  
Public Class PageUpdateLockException  
    Inherits Exception
```

C++


```
[SerializableAttribute]  
public ref class PageUpdateLockException : public Exception
```

F#

```
[<SerializableAttribute>]  
type PageUpdateLockException =  
    class  
        inherit Exception  
    end
```

The **PageUpdateLockException** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------------------------|
|  | PageUpdateLockException | Initializes a new instance of the PageUpdateLockException class |

See Also

[VelocityDb.Exceptions Namespace](#)

PageUpdateLockException Constructor

Initializes a new instance of the [PageUpdateLockException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PageUpdateLockException(  
    string message  
)
```

VB

```
Public Sub New (  
    message As String  
)
```

C++

```
public:  
PageUpdateLockException(  
    String^ message  
)
```

F#

```
new :  
    message : string -> PageUpdateLockException
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for "M:VelocityDb.Exceptions.PageUpdateLockException.#ctor(System.String)"]

See Also

[PageUpdateLockException Class](#)

[VelocityDb.Exceptions Namespace](#)

PersistedObjectExpectedException Class

Exception thrown when a persisted object was expected but was not.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.PersistedObjectExpectedException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class PersistedObjectExpectedException : Exception
```

VB

```
<SerializableAttribute>
Public Class PersistedObjectExpectedException
    Inherits Exception
```

C++


```
[SerializableAttribute]
public ref class PersistedObjectExpectedException : public Exception
```

F#

```
[<SerializableAttribute>]
type PersistedObjectExpectedException =
    class
        inherit Exception
    end
```

The **PersistedObjectExpectedException** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------|
|  | PersistedObjectExpectedException | Initializes a new instance of the PersistedObjectExpectedException class |

See Also

[VelocityDb.Exceptions Namespace](#)

PersistedObjectExpectedException Constructor

Initializes a new instance of the [PersistedObjectExpectedException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public PersistedObjectExpectedException(  
    string message  
)
```

```
VB  
Public Sub New (  
    message As String  
)
```

```
C++  
public:  
PersistedObjectExpectedException(  
    String^ message  
)
```

```
F#  
new :  
    message : string -> PersistedObjectExpectedException
```

Parameters

message

Type: [System.String](#)

[Missing <param name="message"/> documentation for "M:VelocityDb.Exceptions.PersistedObjectExpectedException.#ctor(System.String)"]

See Also

[PersistedObjectExpectedException Class](#)

[VelocityDb.Exceptions Namespace](#)

ReferentialIntegrityException Class

Exception thrown when an internal error was detected. This type of error is most likely due to a bug in VelocityDb.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.ReferentialIntegrityException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class ReferentialIntegrityException : Exception
```

VB

```
<SerializableAttribute>  
Public Class ReferentialIntegrityException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class ReferentialIntegrityException : public Exception
```

F#

```
[<SerializableAttribute>]  
type ReferentialIntegrityException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

RequestedPlacementDatabaseNumberNotValidException Class

Exception thrown when a requested placement Database number is invalid.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.RequestedPlacementDatabaseNumberNotValidException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class RequestedPlacementDatabaseNumberNotValidException : Exception
```

VB

```
<SerializableAttribute>  
Public Class RequestedPlacementDatabaseNumberNotValidException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class RequestedPlacementDatabaseNumberNotValidException : public  
Exception
```

F#

```
[<SerializableAttribute>]  
type RequestedPlacementDatabaseNumberNotValidException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

RequestedPlacementPageNumberNotValidException Class

Exception thrown when a requested placement Page number is invalid.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.RequestedPlacementPageNumberNotValidException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class RequestedPlacementPageNumberNotValidException : Exception
```

VB

```
<SerializableAttribute>  
Public Class RequestedPlacementPageNumberNotValidException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class RequestedPlacementPageNumberNotValidException : public  
Exception
```

F#

```
[<SerializableAttribute>]  
type RequestedPlacementPageNumberNotValidException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

SubscriptionsNotAvailableWithNoServerSessionException Class

Exception thrown when trying to use event subscription api with NON server session. Subscriptions require use of server.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.SubscriptionsNotAvailableWithNoServerSessionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SubscriptionsNotAvailableWithNoServerSessionException :  
Exception
```

VB

```
<SerializableAttribute>  
Public Class SubscriptionsNotAvailableWithNoServerSessionException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class SubscriptionsNotAvailableWithNoServerSessionException :  
public Exception
```

F#

```
[<SerializableAttribute>]  
type SubscriptionsNotAvailableWithNoServerSessionException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

SystemDatabaseNotFoundWithReadOnlyTransactionException Class

Exception thrown when a read only transaction try to open a non existing Database 0

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.SystemDatabaseNotFoundWithReadOnlyTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class SystemDatabaseNotFoundWithReadOnlyTransactionException :  
Exception
```

VB

```
<SerializableAttribute>  
Public Class SystemDatabaseNotFoundWithReadOnlyTransactionException  
    Inherits Exception
```

C++


```
[SerializableAttribute]  
public ref class SystemDatabaseNotFoundWithReadOnlyTransactionException :  
public Exception
```

F#

```
[<SerializableAttribute>]  
type SystemDatabaseNotFoundWithReadOnlyTransactionException =  
    class  
        inherit Exception  
    end
```

The **SystemDatabaseNotFoundWithReadOnlyTransactionException** type exposes the following members.

Constructors

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
|  SystemDatabaseNotFoundWithReadOnlyTransactionException | Use for unexpected events |

See Also

[VelocityDb.Exceptions Namespace](#)

SystemDatabaseNotFoundWithReadOnlyTransactionException Constructor

Use for unexpected events

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SystemDatabaseNotFoundWithReadOnlyTransactionException (  
    string message  
)
```

VB

```
Public Sub New (  
    message As String  
)
```

C++

```
public:  
SystemDatabaseNotFoundWithReadOnlyTransactionException (  
    String^ message  
)
```

F#

```
new :  
    message : string ->  
SystemDatabaseNotFoundWithReadOnlyTransactionException
```

Parameters

message

Type: [System.String](#)

Describe the unexpected event

See Also

[SystemDatabaseNotFoundWithReadOnlyTransactionException Class](#)

[VelocityDb.Exceptions Namespace](#)

TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException Class

Exception thrown when attempting to start a read only transaction while in an update transaction.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class
TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException : Exception
```

VB

```
<SerializableAttribute>
Public Class
TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException
    Inherits Exception
```

C++

```
[SerializableAttribute]
public ref class
TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException : public
Exception
```

F#

```
[<SerializableAttribute>]
type TryingToBeginReadOnlyTransactionWhileInUpdateTransactionException =
    class
        inherit Exception
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

TryingToDeleteDeletedDatabaseException Class

Exception thrown when attempting to delete a Database that already was deleted in the same transaction

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.TryingToDeleteDeletedDatabaseException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class TryingToDeleteDeletedDatabaseException : Exception
```

VB

```
<SerializableAttribute>  
Public Class TryingToDeleteDeletedDatabaseException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class TryingToDeleteDeletedDatabaseException : public Exception
```

F#

```
[<SerializableAttribute>]  
type TryingToDeleteDeletedDatabaseException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

UnexpectedException Class

Exception thrown when something unexpected happened and no specific exception exist for it.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.UnexpectedException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class UnexpectedException : Exception
```

VB

```
<SerializableAttribute>  
Public Class UnexpectedException  
    Inherits Exception
```



C++

```
[SerializableAttribute]  
public ref class UnexpectedException : public Exception
```

F#

```
[<SerializableAttribute>]  
type UnexpectedException =  
    class  
        inherit Exception  
    end
```

Constructors



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------|
|  | UnexpectedException() | Initializes a new instance of the UnexpectedException class |
|  | UnexpectedException(String) | Use for unexpected events |

See Also

[VelocityDb.Exceptions Namespace](#)

UnexpectedException Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------|
|  | UnexpectedException() | Initializes a new instance of the UnexpectedException class |
|  | UnexpectedException(String) | Use for unexpected events |

See Also

[UnexpectedException Class](#)

[VelocityDb.Exceptions Namespace](#)

UnexpectedException Constructor

Initializes a new instance of the [UnexpectedException](#) class

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public UnexpectedException ()
```

VB

```
Public Sub New
```

C++

```
public:  
UnexpectedException ()
```

F#

```
new : unit -> UnexpectedException
```

See Also

[UnexpectedException Class](#)

[UnexpectedException Overload](#)

[VelocityDb.Exceptions Namespace](#)

UnexpectedException Constructor (String)

Use for unexpected events

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public UnexpectedException(  
    string message  
)
```

VB

```
Public Sub New (  
    message As String  
)
```

C++

```
public:  
UnexpectedException(  
    String^ message  
)
```

F#

```
new :  
    message : string -> UnexpectedException
```

Parameters

message

Type: [System.String](#)

Describe the unexpected event

See Also

[UnexpectedException Class](#)

[UnexpectedException Overload](#)

[VelocityDb.Exceptions Namespace](#)

UniqueConstraintException Class

Exception thrown when attempting to open an in database index without specifying a database.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.UniqueConstraintException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class UniqueConstraintException : Exception
```

VB

```
<SerializableAttribute>  
Public Class UniqueConstraintException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class UniqueConstraintException : public Exception
```

F#

```
[<SerializableAttribute>]  
type UniqueConstraintException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

UpdateLockFailedException Class

Exception thrown when an update lock failed.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.UpdateLockFailedException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class UpdateLockFailedException : Exception
```

VB

```
<SerializableAttribute>  
Public Class UpdateLockFailedException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class UpdateLockFailedException : public Exception
```

F#

```
[<SerializableAttribute>]  
type UpdateLockFailedException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityDb.Exceptions Namespace](#)

WeakIOptimizedPersistableReferenceMustBePersistentException Class

Exception thrown when a persisted object was expected but was not.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityDb.Exceptions.WeakIOptimizedPersistableReferenceMustBePersistentException

Namespace: [VelocityDb.Exceptions](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class WeakIOptimizedPersistableReferenceMustBePersistentException :  
Exception
```

VB

```
<SerializableAttribute>  
Public Class WeakIOptimizedPersistableReferenceMustBePersistentException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class WeakIOptimizedPersistableReferenceMustBePersistentException  
: public Exception
```

F#

```
[<SerializableAttribute>]  
type WeakIOptimizedPersistableReferenceMustBePersistentException =  
    class  
        inherit Exception  
    end
```







See Also

[VelocityDb.Exceptions Namespace](#)

VelocityDb.Indexing Namespace

The `VelocityDB.Indexing` namespace contains classes for the VelocityDB indexing feature

Classes

| Class | Description |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Index | Use only with IOptimizedPersistable See http://www.velocitydb.com/UserGuide.aspx#_Toc430849968 |
|  IndexDescriptor | |
|  Indexes | Tracks all indexes in a federation |
|  IndexStringByHashCode | Sort string field by a computed hash instead of normal string ordering. |
|  OnePerDatabase | Use this attribute if you want each index to contain objects of a single database, this type of index is stored within the same database as its indexed objects. |
|  UniqueConstraint | Unique index values can be required by using the UniqueConstraint attribute |

Index Class

Use only with [IOptimizedPersistable](#) See http://www.velocitydb.com/UserGuide.aspx#_Toc430849968

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.Indexing.Index

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class Index : Attribute
```

VB

```
Public NotInheritable Class Index
    Inherits Attribute
```

C++



```
public ref class Index sealed : public Attribute
```

F#


```
[<SealedAttribute>]
type Index =
    class
        inherit Attribute
    end
```

The **Index** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Index() | Index by default field which is ld |
|  | Index(String) | Create index by one or more field names separated by comma. First field is used first in lookups then when there is multiple matches, field 2 is used and so on. |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|---------------------------------|
|  | FieldNames | Field names used by this index. |



VelocityDB Class Library

See Also

[VelocityDb.Indexing Namespace](#)

Index Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Index() | Index by default field which is Id |
|  | Index(String) | Create index by one or more field names separated by comma. First field is used first in lookups then when there is multiple matches, field 2 is used and so on. |

See Also

[Index Class](#)

[VelocityDb.Indexing Namespace](#)

Index Constructor

Index by default field which is [Id](#)

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Index ()
```

VB

```
Public Sub New
```

C++

```
public:  
Index ()
```

F#

```
new : unit -> Index
```

See Also

[Index Class](#)

[Index Overload](#)

[VelocityDb.Indexing Namespace](#)

Index Constructor (String)

Create index by one or more field names separated by comma. First field is used first in lookups then when there is multiple matches, field 2 is used and so on.

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Index(  
    string fieldNames  
)
```

VB

```
Public Sub New (  
    fieldNames As String  
)
```

C++

```
public:  
Index(  
    String^ fieldNames  
)
```

F#

```
new :  
    fieldNames : string -> Index
```

Parameters

fieldNames

Type: [System.String](#)

Field names in lookup order separated by comma.

See Also

[Index Class](#)


[Index Overload](#)

[VelocityDb.Indexing Namespace](#)

Index.Index Properties

The [Index](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|---------------------------------|
|  | FieldNames | Field names used by this index. |

See Also

[Index Class](#)

[VelocityDb.Indexing Namespace](#)

Index.FieldNames Property

Field names used by this index.

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string FieldNames { get; }
```

VB

```
Public ReadOnly Property FieldNames As String  
    Get
```

C++

```
public:  
property String^ FieldNames {  
    String^ get ();  
}
```

F#

```
member FieldNames : string with get
```

Property Value

Type: [String](#)

See Also

[Index Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor Class

[Missing <summary> documentation for "T:VelocityDb.Indexing.IndexDescriptor"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Indexing.IndexDescriptor

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
[SerializableAttribute]
public class IndexDescriptor : OptimizedPersistable
```



```
VB
<SerializableAttribute>
Public Class IndexDescriptor
    Inherits OptimizedPersistable
```

```
C++
[SerializableAttribute]
public ref class IndexDescriptor : public OptimizedPersistable
```

```
F#
[<SerializableAttribute>]
type IndexDescriptor =
    class
        inherit OptimizedPersistable
    end
```

The **IndexDescriptor** type exposes the following members.





Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------------------|
|  | IndexDescriptor(UInt32, UInt32, UInt32) | Initializes a new instance of the IndexDescriptor class |
|  | IndexDescriptor(UInt32, UInt32, Oid) | Initializes a new instance of the IndexDescriptor class |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-------------|
|  | BTreeld | |

VelocityDB Class Library

| | |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
|  DataMemberShortId | |
|  PlacementDatabaseNumber | (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  ShortRefDatabaseNumber | |
|  TypeShortId | |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Indexing Namespace](#)

IndexDescriptor Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------------------|
|  | IndexDescriptor(UInt32, UInt32, UInt32) | Initializes a new instance of the IndexDescriptor class |
|  | IndexDescriptor(UInt32, UInt32, Oid) | Initializes a new instance of the IndexDescriptor class |

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor Constructor (UInt32, UInt32, UInt32)

Initializes a new instance of the [IndexDescriptor](#) class

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IndexDescriptor(  
    uint typeShortId,  
    uint dataMemberShortId = 0,  
    uint databaseId = 0  
)
```

VB

```
Public Sub New (  
    typeShortId As UInteger,  
    Optional dataMemberShortId As UInteger = 0,  
    Optional databaseId As UInteger = 0  
)
```

C++

```
public:  
IndexDescriptor(  
    unsigned int typeShortId,  
    unsigned int dataMemberShortId = 0,  
    unsigned int databaseId = 0  
)
```

F#

```
new :  
    typeShortId : uint32 *  
    ?dataMemberShortId : uint32 *  
    ?databaseId : uint32  
(* Defaults:  
    let _dataMemberShortId = defaultArg dataMemberShortId 0  
    let _databaseId = defaultArg databaseId 0  
*)  
-> IndexDescriptor
```

Parameters

typeShortId

Type: [System.UInt32](#)

[Missing <param name="typeShortId"/> documentation for "M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,System.UInt32)"]

dataMemberShortId (Optional)

Type: [System.UInt32](#)

[Missing <param name="dataMemberShortId"/> documentation for "M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,System.UInt32)"]

databaseId (Optional)

Type: [System.UInt32](#)

[Missing <param name="databaseId"/> documentation for "M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,System.UInt32)"]

See Also

[IndexDescriptor Class](#)

[IndexDescriptor Overload](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor Constructor (UInt32, UInt32, Oid)

Initializes a new instance of the [IndexDescriptor](#) class

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IndexDescriptor(  
    uint typeShortId,  
    uint dataMemberShortId,  
    Oid bTreeId  
)
```

VB

```
Public Sub New (  
    typeShortId As UInteger,  
    dataMemberShortId As UInteger,  
    bTreeId As Oid  
)
```

C++

```
public:  
IndexDescriptor(  
    unsigned int typeShortId,  
    unsigned int dataMemberShortId,  
    Oid bTreeId  
)
```

F#

```
new :  
    typeShortId : uint32 *  
    dataMemberShortId : uint32 *  
    bTreeId : Oid -> IndexDescriptor
```

Parameters

typeShortId

Type: [System.UInt32](#)

[Missing <param name="typeShortId"/> documentation for "M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,VelocityDb.Oid)"]

dataMemberShortId

Type: [System.UInt32](#)

[Missing <param name="dataMemberShortId"/> documentation for "M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,VelocityDb.Oid)"]

bTreeId

Type: [VelocityDb.Oid](#)

**[Missing <param name="bTreeld"/> documentation for
"M:VelocityDb.Indexing.IndexDescriptor.#ctor(System.UInt32,System.UInt32,VelocityDb.Oid)"]**

See Also

[IndexDescriptor Class](#)






[IndexDescriptor Overload](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.IndexDescriptor Properties

The [IndexDescriptor](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------------------------------------------|
|  | BTreeld | |
|  | DataMemberShortId | |
|  | PlacementDatabaseNumber | (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |
|  | ShortRefDatabaseNumber | |
|  | TypeShortId | |

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.BTreeId Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.IndexDescriptor.BTreeId"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid BTreeId { get; set; }
```

VB

```
Public Property BTreeId As Oid  
    Get  
    Set
```

C++

```
public:  
property Oid BTreeId {  
    Oid get ();  
    void set (Oid value);  
}
```

F#

```
member BTreeId : Oid with get, set
```

Property Value

Type: [Oid](#)

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.DataMemberShortId Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.IndexDescriptor.DataMemberShortId"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint DataMemberShortId { get; }
```

VB

```
Public ReadOnly Property DataMemberShortId As UInteger  
    Get
```

C++

```
public:  
property unsigned int DataMemberShortId {  
    unsigned int get ();  
}
```

F#

```
member DataMemberShortId : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.PlacementDatabaseNumber Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.IndexDescriptor.PlacementDatabaseNumber"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint PlacementDatabaseNumber { get; }
```

VB

```
Public Overrides ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int PlacementDatabaseNumber {  
    unsigned int get () override;  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get  
override PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.PlacementDatabaseNumber](#)

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.ShortRefDatabaseNumber Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.IndexDescriptor.ShortRefDatabaseNumber"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint ShortRefDatabaseNumber { get; }
```

VB

```
Public ReadOnly Property ShortRefDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
property unsigned int ShortRefDatabaseNumber {  
    unsigned int get ();  
}
```

F#

```
member ShortRefDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.TypeShortId Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.IndexDescriptor.TypeShortId"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint TypeShortId { get; }
```

VB

```
Public ReadOnly Property TypeShortId As UInteger  
    Get
```

C++

```
public:  
property unsigned int TypeShortId {  
    unsigned int get ();  
}
```

F#

```
member TypeShortId : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also



[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

IndexDescriptor.IndexDescriptor Methods

The [IndexDescriptor](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[IndexDescriptor Class](#)

[VelocityDb.Indexing Namespace](#)

Indexes Class

Tracks all indexes in a federation

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Indexing.Indexes

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
[SerializableAttribute]
public class Indexes : OptimizedPersistable
```


```
VB
<SerializableAttribute>
Public Class Indexes
    Inherits OptimizedPersistable
```

```
C++
[SerializableAttribute]
public ref class Indexes : public OptimizedPersistable
```



```
F#
[<SerializableAttribute>]
type Indexes =
    class
        inherit OptimizedPersistable
    end
```

The **Indexes** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------|
|  | Indexes | Initializes a new instance of the Indexes class |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
|  | IndexDescriptors | |
|  | PlacementDatabaseNumber | Hint about where to persist DatabaseLocation (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Open | |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Indexing Namespace](#)

Indexes Constructor

Initializes a new instance of the [Indexes](#) class

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Indexes (  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    session As SessionBase  
)
```

C++

```
public:  
Indexes (  
    SessionBase^ session  
)
```

F#

```
new :  
    session : SessionBase -> Indexes
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.Indexing.Indexes.#ctor(VelocityDb.Session.SessionBase)"]

See Also



[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)

Indexes.Indexes Properties

The [Indexes](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
|  | IndexDescriptors | |
|  | PlacementDatabaseNumber | Hint about where to persist DatabaseLocation (Overrides OptimizedPersistable.PlacementDatabaseNumber.) |

See Also

[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)

Indexes.IndexDescriptors Property

[Missing <summary> documentation for "P:VelocityDb.Indexing.Indexes.IndexDescriptors"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSetOidShort<IndexDescriptor> IndexDescriptors { get; }
```

VB

```
Public ReadOnly Property IndexDescriptors As BTreeSetOidShort(Of  
IndexDescriptor)  
    Get
```

C++

```
public:  
property BTreeSetOidShort<IndexDescriptor^>^ IndexDescriptors {  
    BTreeSetOidShort<IndexDescriptor^>^ get ();  
}
```

F#

```
member IndexDescriptors : BTreeSetOidShort<IndexDescriptor> with get
```

Property Value

Type: [BTreeSetOidShort\(IndexDescriptor\)](#)

See Also

[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)

Indexes.PlacementDatabaseNumber Property

Hint about where to persist DatabaseLocation

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override uint PlacementDatabaseNumber { get; }
```

VB

```
Public Overrides ReadOnly Property PlacementDatabaseNumber As UInteger  
    Get
```

C++

```
public:  
virtual property unsigned int PlacementDatabaseNumber {  
    unsigned int get () override;  
}
```

F#

```
abstract PlacementDatabaseNumber : uint32 with get  
override PlacementDatabaseNumber : uint32 with get
```

Property Value

Type: [UInt32](#)

Implements

[IOptimizedPersistable.PlacementDatabaseNumber](#)

See Also


[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)



Indexes.Indexes Methods

The [Indexes](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Open | |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)

Indexes.Open Method

[Missing <summary> documentation for

"M:VelocityDb.Indexing.Indexes.Open(VelocityDb.Session.SessionBase,System.Boolean)"]

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Indexes Open(  
    SessionBase session,  
    bool update  
)
```

VB

```
Public Shared Function Open (  
    session As SessionBase,  
    update As Boolean  
) As Indexes
```

C++

```
public:  
static Indexes^ Open(  
    SessionBase^ session,  
    bool update  
)
```

F#

```
static member Open :  
    session : SessionBase *  
    update : bool -> Indexes
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.Indexing.Indexes.Open(VelocityDb.Session.SessionBase,System.Boolean)"]

update

Type: [System.Boolean](#)

[Missing <param name="update"/> documentation for

"M:VelocityDb.Indexing.Indexes.Open(VelocityDb.Session.SessionBase,System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [Indexes](#)

[Missing <returns> documentation for

"M:VelocityDb.Indexing.Indexes.Open(VelocityDb.Session.SessionBase,System.Boolean)"]

See Also

[Indexes Class](#)

[VelocityDb.Indexing Namespace](#)

IndexStringByHashCode Class

Sort string field by a computed hash instead of normal string ordering.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.Indexing.IndexStringByHashCode

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class IndexStringByHashCode : Attribute
```

VB

```
Public NotInheritable Class IndexStringByHashCode  
    Inherits Attribute
```

C++


```
public ref class IndexStringByHashCode sealed : public Attribute
```

F#

```
[<SealedAttribute>]  
type IndexStringByHashCode =  
    class  
        inherit Attribute  
    end
```

The **IndexStringByHashCode** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------|
|  | IndexStringByHashCode | Initializes a new instance of the IndexStringByHashCode class |

See Also

[VelocityDb.Indexing Namespace](#)

IndexStringByHashCode Constructor

Initializes a new instance of the [IndexStringByHashCode](#) class

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IndexStringByHashCode ()
```

VB

```
Public Sub New
```

C++

```
public:  
IndexStringByHashCode ()
```

F#

```
new : unit -> IndexStringByHashCode
```

See Also

[IndexStringByHashCode Class](#)

[VelocityDb.Indexing Namespace](#)

OnePerDatabase Class

Use this attribute if you want each index to contain objects of a single database, this type of index is stored within the same database as its indexed objects.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.Indexing.OnePerDatabase

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class OnePerDatabase : Attribute
```

VB

```
Public NotInheritable Class OnePerDatabase  
    Inherits Attribute
```

C++


```
public ref class OnePerDatabase sealed : public Attribute
```

F#

```
[<SealedAttribute>]  
type OnePerDatabase =  
    class  
        inherit Attribute  
    end
```

The **OnePerDatabase** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | OnePerDatabase | Initializes a new instance of the OnePerDatabase class |

See Also

[VelocityDb.Indexing Namespace](#)

OnePerDatabase Constructor

Initializes a new instance of the [OnePerDatabase](#) class

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OnePerDatabase ()
```

VB

```
Public Sub New
```

C++

```
public:  
OnePerDatabase ()
```

F#

```
new : unit -> OnePerDatabase
```

See Also

[OnePerDatabase Class](#)

[VelocityDb.Indexing Namespace](#)

UniqueConstraint Class

Unique index values can be required by using the UniqueConstraint attribute

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.Indexing.UniqueConstraint

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class UniqueConstraint : Attribute
```

VB

```
Public NotInheritable Class UniqueConstraint  
    Inherits Attribute
```

C++

```
public ref class UniqueConstraint sealed : public Attribute
```

F#

```
[<SealedAttribute>]  
type UniqueConstraint =  
    class  
        inherit Attribute  
    end
```

The **UniqueConstraint** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | UniqueConstraint | Constructor |

See Also

[VelocityDb.Indexing Namespace](#)

UniqueConstraint Constructor

Constructor

Namespace: [VelocityDb.Indexing](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public UniqueConstraint ()
```

VB

```
Public Sub New
```

C++

```
public:  
UniqueConstraint ()
```

F#

```
new : unit -> UniqueConstraint
```

See Also










[UniqueConstraint Class](#)

[VelocityDb.Indexing Namespace](#)

VelocityDb.Session Namespace

The `VelocityDb.Session` namespace contains session classes for interfacing with VelocityDb

Classes

| Class | Description |
|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReplicaInfo | Specifies a location for a replica of primary DatabaseLocation containing at least the system databases (0.oddb, 1.oddb, 2.oddb ...) |
|  ServerClientSession | Use this session class when Page level locking is required and/or not all Databases are local and not reachable by UNC paths. |
|  ServerClientSessionShared | Use as a shared session for multiple reader threads connected to one or more VelocityDBServer(s). |
|  SessionBase | Transaction control, database creation and special persistent object api. Each session can use 4,294,967,295 Databases . Each Database may have up to 65,535 Pages and a page can be up to 2 GB in size (.Net restriction of a byte array). The maximum data size a single session can manage is: $4,294,967,295 * 65,535 * 2147483648 = 604453686294542391705600$ bytes or about half a Yottabyte http://en.wikipedia.org/wiki/Yottabyte |
|  SessionBase.Transaction | Helper object to enable automatic rollback/abort of transaction if transaction isn't committed within it's scope. |
|  SessionNoServer | Use this session class when Database level locking is acceptable and Databases are local or reachable by UNC paths. |
|  SessionNoServerShared | Same as SessionNoServer but SessionNoServerShared is thread safe so it can be shared between threads. |
|  SessionPool | Provides a pool of sessions for reuse instead of creating new session instances every time a session is needed. The pooled sessions includes cached databases, pages and objects. Keep the pool size to a minimum too avoid excessive memory usage. |
|  TypeExtensions | Some extensions to Type |

ReplicaInfo Class

Specifies a location for a replica of primary [DatabaseLocation](#) containing at least the system databases (0.odbc, 1.odbc, 2.odbc ...)

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Session.ReplicaInfo

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReplicaInfo : IComparable<ReplicaInfo>
```

VB

```
Public Class ReplicaInfo
    Implements IComparable(Of ReplicaInfo)
```

C++


```
public ref class ReplicaInfo : IComparable<ReplicaInfo^>
```

F#



```
type ReplicaInfo =
    class
        interface IComparable<ReplicaInfo>
    end
```

The **ReplicaInfo** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------|
|  | ReplicaInfo | Initializes a new instance of the ReplicaInfo class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|------------------------------------|
|  | Host | Host name (defaults to local host) |
|  | Path | Directory path to databases |

See Also

[VelocityDb.Session Namespace](#)

ReplicaInfo Constructor

Initializes a new instance of the [ReplicaInfo](#) class

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReplicaInfo ()
```

VB

```
Public Sub New
```

C++

```
public:  
ReplicaInfo ()
```

F#

```
new : unit -> ReplicaInfo
```

See Also



[ReplicaInfo Class](#)

[VelocityDb.Session Namespace](#)

ReplicaInfo.ReplicaInfo Properties

The [ReplicaInfo](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|------------------------------------|
|  | Host | Host name (defaults to local host) |
|  | Path | Directory path to databases |

See Also

[ReplicaInfo Class](#)

[VelocityDb.Session Namespace](#)

ReplicaInfo.Host Property

Host name (defaults to local host)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Host { get; set; }
```

VB

```
Public Property Host As String  
    Get  
    Set
```

C++

```
public:  
property String^ Host {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member Host : string with get, set
```

Property Value

Type: [String](#)

See Also

[ReplicaInfo Class](#)

[VelocityDb.Session Namespace](#)

ReplicaInfo.Path Property

Directory path to databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Path { get; set; }
```

VB

```
Public Property Path As String  
    Get  
    Set
```

C++

```
public:  
property String^ Path {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member Path : string with get, set
```

Property Value

Type: [String](#)

See Also

[ReplicaInfo Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession Class

Use this session class when [Page](#) level locking is required and/or not all [Databases](#) are local and not reachable by UNC paths.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

VelocityDb.Session.ServerClientSession

[VelocityDb.Session.ServerClientSessionShared](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ServerClientSession : SessionBase
```

VB

```
Public Class ServerClientSession
    Inherits SessionBase
```

C++



```
public ref class ServerClientSession : public SessionBase
```

F#



```
type ServerClientSession =
    class
        inherit SessionBase
    end
```

The **ServerClientSession** type exposes the following members.














Constructors










| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
|  ServerClientSession(List(ReplicaInfo), Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean) | Initializes a new instance of the ServerClientSession class |
|  ServerClientSession(String, String, Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean) | Creates a new session. |

Properties








| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddToIndexInSeperateThread | Indexing thread not yet available with ServerClientSession . (Overrides SessionBase.AddToIndexInSeperateThread .) |
|  WriteToDiskInSeperateDatabaseThreads | Write threads not yet available with ServerClientSession . (Overrides SessionBase.WriteToDiskInSeperateDatabaseThreads .) |

Methods



| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Aborts a transaction, it undoes all persistent changes made within the transaction (Overrides SessionBase.Abort() .) |
|  BeginRead | Transaction control, begin a read only transaction (Overrides SessionBase.BeginRead(Boolean) .) |
|  BeginReadWithEvents | Start a read only transaction and gets a list of created/updated objects as subscribed to by this session |
|  BeginUpdate | Transaction control, begin an update transaction. (Overrides SessionBase.BeginUpdate(Boolean) .) |
|  BeginUpdateWithEvents | Starts a transaction that may be creating new persistent data and/or updating persistent data and gets a list of created/updated objects as subscribed to by this session |
|  ClearServerCache | Used as an aid when debugging server data, internal use, avoid using for now. |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact() .) |
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database) .) |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides SessionBase.CopyAllDatabasesTo(String, Boolean) .) |
|  DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides SessionBase.DeleteLocation(DatabaseLocation, Boolean) .) |
|  FlushUpdates | Send all updated pages to the server(s) managing those pages, freeing up memory in client. (Overrides SessionBase.FlushUpdates() .) |
|  FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) (Overrides SessionBase.FlushUpdatesServers() .) |
|  InUseNumberOfPages | Gets the number of pages currently in use by a Database (Overrides SessionBase.InUseNumberOfPages(Database) .) |

| | |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  NewDatabase | Create a new Database with a given database number (Overrides SessionBase.NewDatabase(UInt32, UInt32, String, Boolean).) |
|  NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides SessionBase.NewLocation(DatabaseLocation).) |
|  OpenAllDatabases | Open all databases (Overrides SessionBase.OpenAllDatabases(Boolean).) |
|  OpenDatabase | Opens a Database (Overrides SessionBase.OpenDatabase(UInt32, Boolean, Boolean).) |
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides SessionBase.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides SessionBase.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.SubscribeToChanges(Type, String).) |
|  UnsubscribeToChanges | Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.UnsubscribeToChanges(Type, String).) |
|  UpdateDatabase | Request an update lock on a database (Overrides SessionBase.UpdateDatabase(Database).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |

VelocityDB Class Library



| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[VelocityDb.Session Namespace](#)

ServerClientSession Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  | ServerClientSession(List(ReplicaInfo), Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean) | Initializes a new instance of the ServerClientSession class |
|  | ServerClientSession(String, String, Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean) | Creates a new session. |

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession Constructor (List(ReplicaInfo), Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean)

Initializes a new instance of the [ServerClientSession](#) class

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ServerClientSession(
    List<ReplicaInfo> systemHostPath,
    int waitForLockMilliseconds = 2000,
    bool optimisticLocking = true,
    bool inMemoryOnly = false,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes,
    bool enableDatabaseCache = true
)
```

VB

```
Public Sub New (
    systemHostPath As List(Of ReplicaInfo),
    Optional waitForLockMilliseconds As Integer = 2000,
    Optional optimisticLocking As Boolean = true,
    Optional inMemoryOnly As Boolean = false,
    Optional enablePageCache As Boolean = true,
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes,
    Optional enableDatabaseCache As Boolean = true
)
```

C++

```
public:
ServerClientSession(
    List<ReplicaInfo>^ systemHostPath,
    int waitForLockMilliseconds = 2000,
    bool optimisticLocking = true,
    bool inMemoryOnly = false,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes,
    bool enableDatabaseCache = true
)
```

F#

```
new :
    systemHostPath : List<ReplicaInfo> *
    ?waitForLockMilliseconds : int *
    ?optimisticLocking : bool *
    ?inMemoryOnly : bool *
    ?enablePageCache : bool *
```

```
?objectCachingDefaultPolicy : CacheEnum *
?enableDatabaseCache : bool
(* Defaults:
    let _waitForLockMilliseconds = defaultArg waitForLockMilliseconds 2000
    let _optimisticLocking = defaultArg optimisticLocking true
    let _inMemoryOnly = defaultArg inMemoryOnly false
    let _enablePageCache = defaultArg enablePageCache true
    let _objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
    let _enableDatabaseCache = defaultArg enableDatabaseCache true
*)
-> ServerClientSession
```

Parameters

systemHostPath

Type: [System.Collections.Generic.List<ReplicaInfo>](#)

[Missing <param name="systemHostPath"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

[Missing <param name="waitForLockMilliseconds"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

optimisticLocking (Optional)

Type: [System.Boolean](#)

[Missing <param name="optimisticLocking"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

inMemoryOnly (Optional)

Type: [System.Boolean](#)

[Missing <param name="inMemoryOnly"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

enablePageCache (Optional)

Type: [System.Boolean](#)

[Missing <param name="enablePageCache"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

[Missing <param name="objectCachingDefaultPolicy"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

enableDatabaseCache (Optional)

Type: [System.Boolean](#)

[Missing <param name="enableDatabaseCache"/> documentation for "M:VelocityDb.Session.ServerClientSession.#ctor(System.Collections.Generic.List{VelocityDb.Session.ReplicaInfo},System.Int32,System.Boolean,System.Boolean,System.Boolean,VelocityDb.CacheEnum,System.Boolean)"]

See Also

[ServerClientSession Class](#)

[ServerClientSession Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSession Constructor (String, String, Int32, Boolean, Boolean, Boolean, CacheEnum, Boolean)

Creates a new session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ServerClientSession(  
    string systemDir,  
    string systemHost = null,  
    int waitForLockMilliseconds = 2000,  
    bool optimisticLocking = true,  
    bool inMemoryOnly = false,  
    bool enablePageCache = true,  
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes,  
    bool enableDatabaseCache = true  
)
```

VB

```
Public Sub New (  
    systemDir As String,  
    Optional systemHost As String = Nothing,  
    Optional waitForLockMilliseconds As Integer = 2000,  
    Optional optimisticLocking As Boolean = true,  
    Optional inMemoryOnly As Boolean = false,  
    Optional enablePageCache As Boolean = true,  
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes,  
    Optional enableDatabaseCache As Boolean = true  
)
```

C++

```
public:  
ServerClientSession(  
    String^ systemDir,  
    String^ systemHost = nullptr,  
    int waitForLockMilliseconds = 2000,  
    bool optimisticLocking = true,  
    bool inMemoryOnly = false,  
    bool enablePageCache = true,  
    CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes,  
    bool enableDatabaseCache = true  
)
```

F#

```
new :  
    systemDir : string *  
    ?systemHost : string *
```



```
?waitForLockMilliseconds : int *
?optimisticLocking : bool *
?inMemoryOnly : bool *
?enablePageCache : bool *
?objectCachingDefaultPolicy : CacheEnum *
?enableDatabaseCache : bool
(* Defaults:
  let_systemHost = defaultArg systemHost null
  let_waitForLockMilliseconds = defaultArg waitForLockMilliseconds 2000
  let_optimisticLocking = defaultArg optimisticLocking true
  let_inMemoryOnly = defaultArg inMemoryOnly false
  let_enablePageCache = defaultArg enablePageCache true
  let_objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
  let_enableDatabaseCache = defaultArg enableDatabaseCache true
*)
-> ServerClientSession
```

Parameters

systemDir

Type: [System.String](#)

The startup location directory path (do not use UNC path). Use path as it is on the host owning the directory. Path can be an absolute (full) path or a path relative to **s_baseDatabasePath**. Change default of **s_baseDatabasePath** on server by starting server with a forth parameter specifying the base path

systemHost (Optional)

Type: [System.String](#)

Hostname of the host owning the systemDir

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

The desired maximum lock wait time

optimisticLocking (Optional)

Type: [System.Boolean](#)

Use optimistic locking. With optimistic locking, readers are always permitted and multiple updaters are permitted to update the same Databases/Pages but only the first transaction to commit or flush an updated Database/Page will be successful in making the commit change. Other updaters will get an `OptimisticLockingException` if they try to commit/flush updates to the same Pages/Database. See http://en.wikipedia.org/wiki/Optimistic_concurrency_control for further explanation of optimistic locking. If optimistic locking is not enabled then pessimistic locking is used. See: http://en.wikipedia.org/wiki/Concurrency_control

inMemoryOnly (Optional)

Type: [System.Boolean](#)

Disable all disk access by using this option. Best if used in combination with backup [DatabaseLocation](#). See High availability sample application

enablePageCache (Optional)

Type: [System.Boolean](#)

VelocityDB Class Library

If strong reference page caching isn't desired, turn it off by setting to false

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

Determines default value of [Cache](#)

enableDatabaseCache (Optional)

Type: [System.Boolean](#)

Is this session going to enable strong reference [Database](#) caching?

See Also

[ServerClientSession Class](#)



[ServerClientSession Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.ServerClientSession Properties

The [ServerClientSession](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddToIndexInSeperateThread | Indexing thread not yet available with ServerClientSession . (Overrides SessionBase.AddToIndexInSeperateThread .) |
|  | WriteToDiskInSeperateDatabaseThreads | Write threads not yet available with ServerClientSession . (Overrides SessionBase.WriteToDiskInSeperateDatabaseThreads .) |

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.AddToIndexInSeperateThread Property

Indexing thread not yet available with [ServerClientSession](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AddToIndexInSeperateThread { get; set; }
```

VB

```
Public Overrides Property AddToIndexInSeperateThread As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool AddToIndexInSeperateThread {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract AddToIndexInSeperateThread : bool with get, set  
override AddToIndexInSeperateThread : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.WriteToDiskInSeperateDatabaseThreads Property

Write threads not yet available with [ServerClientSession](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool WriteToDiskInSeperateDatabaseThreads { get; set; }
```

VB

```
Public Overrides Property WriteToDiskInSeperateDatabaseThreads As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool WriteToDiskInSeperateDatabaseThreads {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract WriteToDiskInSeperateDatabaseThreads : bool with get, set  
override WriteToDiskInSeperateDatabaseThreads : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

















[ServerClientSession Class](#)







[VelocityDb.Session Namespace](#)

ServerClientSession.ServerClientSession Methods










The [ServerClientSession](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Aborts a transaction, it undoes all persistent changes made within the transaction (Overrides SessionBase.Abort().) |
|  BeginRead | Transaction control, begin a read only transaction (Overrides SessionBase.BeginRead(Boolean).) |
|  BeginReadWithEvents | Start a read only transaction and gets a list of created/updated objects as subscribed to by this session |
|  BeginUpdate | Transaction control, begin an update transaction. (Overrides SessionBase.BeginUpdate(Boolean).) |
|  BeginUpdateWithEvents | Starts a transaction that may be creating new persistent data and/or updating persistent data and gets a list of created/updated objects as subscribed to by this session |
|  ClearServerCache | Used as an aid when debugging server data, internal use, avoid using for now. |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact().) |
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database).) |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides SessionBase.CopyAllDatabasesTo(String, Boolean).) |
|  DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides SessionBase.DeleteLocation(DatabaseLocation, Boolean).) |
|  FlushUpdates | Send all updated pages to the server(s) managing those pages, freeing up memory in client. (Overrides SessionBase.FlushUpdates().) |
|  FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) (Overrides SessionBase.FlushUpdatesServers().) |
|  InUseNumberOfPages | Gets the number of pages currently in use by a Database (Overrides SessionBase.InUseNumberOfPages(Database).) |
|  NewDatabase | Create a new Database with a given database number (Overrides SessionBase.NewDatabase(UInt32, UInt32, String, Boolean).) |
|  NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides SessionBase.NewLocation(DatabaseLocation).) |
|  OpenAllDatabases | Open all databases (Overrides SessionBase.OpenAllDatabases(Boolean).) |

| | |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OpenDatabase | Opens a Database (Overrides SessionBase.OpenDatabase(UInt32, Boolean, Boolean).) |
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides SessionBase.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides SessionBase.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.SubscribeToChanges(Type, String).) |
|  UnsubscribeToChanges | Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.UnsubscribeToChanges(Type, String).) |
|  UpdateDatabase | Request an update lock on a database (Overrides SessionBase.UpdateDatabase(Database).) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

VelocityDB Class Library

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.Abort Method

Aborts a transaction, it undoes all persistent changes made within the transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Abort ()
```

VB

```
Public Overrides Sub Abort
```

C++

```
public:  
virtual void Abort () override
```

F#

```
abstract Abort : unit -> unit  
override Abort : unit -> unit
```

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.BeginRead Method

Transaction control, begin a read only transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginRead(  
    bool doRecoveryCheck = false  
)
```

VB

```
Public Overrides Function BeginRead (  
    Optional doRecoveryCheck As Boolean = false  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginRead(  
    bool doRecoveryCheck = false  
) override
```

F#

```
abstract BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction  
override BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

By default we do not check for a failed update transaction that requires reverting one or more database changes. We don't do it by default because it can be a little costly. Change this parameter to `true` if you prefer ensured consistency over better performance or make change only if you encounter an exception in the transaction.

VelocityDB Class Library

Return Value

Type: [SessionBase.Transaction](#)

A transaction helper object.

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.BeginReadWithEvents Method

Start a read only transaction and gets a list of created/updated objects as subscribed to by this session

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<Oid> BeginReadWithEvents (  
    bool doRecoveryCheck = true  
)
```

VB

```
Public Function BeginReadWithEvents (  
    Optional doRecoveryCheck As Boolean = true  
) As List(Of Oid)
```

C++

```
public:  
List<Oid>^ BeginReadWithEvents (  
    bool doRecoveryCheck = true  
)
```

F#

```
member BeginReadWithEvents :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck true  
)  
-> List<Oid>
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

If true a recovery check is done, otherwise recovery check is bypassed.

Return Value

Type: [List\(Oid\)](#)

list of object Oid of objects updated by other sessions

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.BeginUpdate Method

Transaction control, begin an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginUpdate(  
    bool doRecoveryCheck = true  
)
```

VB

```
Public Overrides Function BeginUpdate (  
    Optional doRecoveryCheck As Boolean = true  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginUpdate(  
    bool doRecoveryCheck = true  
) override
```

F#

```
abstract BeginUpdate :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck true  
)  
-> SessionBase.Transaction  
override BeginUpdate :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck true  
)  
-> SessionBase.Transaction
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

Set to false only when moving system databases to a new directory

Return Value

Type: [SessionBase.Transaction](#)

A [SessionBase.Transaction](#) helper object.

See Also

[ServerClientSession Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

ServerClientSession.BeginUpdateWithEvents Method

Starts a transaction that may be creating new persistent data and/or updating persistent data and gets a list of created/updated objects as subscribed to by this session

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<Oid> BeginUpdateWithEvents (  
    bool doRecoveryCheck = true  
)
```

VB

```
Public Function BeginUpdateWithEvents (  
    Optional doRecoveryCheck As Boolean = true  
) As List(Of Oid)
```

C++

```
public:  
List<Oid>^ BeginUpdateWithEvents (  
    bool doRecoveryCheck = true  
)
```

F#

```
member BeginUpdateWithEvents :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let doRecoveryCheck = defaultArg doRecoveryCheck true  
)  
-> List<Oid>
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

If true a recovery check is done, otherwise recovery check is bypassed.

Return Value

Type: [List\(Oid\)](#)

list of object Oid of objects updated by other sessions

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.ClearServerCache Method

Used as an aid when debugging server data, internal use, avoid using for now.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ClearServerCache ()
```

VB

```
Public Sub ClearServerCache
```

C++

```
public:  
void ClearServerCache ()
```

F#

```
member ClearServerCache : unit -> unit
```



See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.Compact Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact().) |
|  | Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database).) |

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.Compact Method

Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each [Database](#) file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact ()
```

VB

```
Public Overrides Sub Compact
```

C++

```
public:  
virtual void Compact () override
```

F#

```
abstract Compact : unit -> unit  
override Compact : unit -> unit
```

See Also

[ServerClientSession Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.Compact Method (Database)

Reduce size of database, if possible, by truncating file where unused space begins

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact (  
    Database db  
)
```

VB

```
Public Overrides Sub Compact (  
    db As Database  
)
```

C++

```
public:  
virtual void Compact (  
    Database^ db  
) override
```

F#

```
abstract Compact :  
    db : Database -> unit  
override Compact :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

to compact

See Also

[ServerClientSession Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.CopyAllDatabasesTo Method

Copies all databases to a selected directory on the local host.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void CopyAllDatabasesTo(  
    string directory,  
    bool systemDatabaseLocationOnly = false  
)
```

VB

```
Public Overrides Sub CopyAllDatabasesTo (  
    directory As String,  
    Optional systemDatabaseLocationOnly As Boolean = false  
)
```

C++

```
public:  
virtual void CopyAllDatabasesTo(  
    String^ directory,  
    bool systemDatabaseLocationOnly = false  
) override
```

F#

```
abstract CopyAllDatabasesTo :  
    directory : string *  
    ?systemDatabaseLocationOnly : bool  
(* Defaults:  
    let_systemDatabaseLocationOnly = defaultArg  
systemDatabaseLocationOnly false  
)  
-> unit  
override CopyAllDatabasesTo :  
    directory : string *  
    ?systemDatabaseLocationOnly : bool  
(* Defaults:  
    let_systemDatabaseLocationOnly = defaultArg  
systemDatabaseLocationOnly false  
)  
-> unit
```

Parameters

directory

Type: [System.String](#)

Path to a directory

VelocityDB Class Library

systemDatabaseLocationOnly (Optional)

Type: [System.Boolean](#)

Only copy databases in system database location

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.DeleteLocation Method

Deletes a [DatabaseLocation](#), location must first not have any Databases in it

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void DeleteLocation(
    DatabaseLocation location,
    bool catalogOnly = false
)
```

VB

```
Public Overrides Sub DeleteLocation (
    location As DatabaseLocation,
    Optional catalogOnly As Boolean = false
)
```

C++

```
public:
    virtual void DeleteLocation(
        DatabaseLocation^ location,
        bool catalogOnly = false
    ) override
```

F#

```
abstract DeleteLocation :
    location : DatabaseLocation *
    ?catalogOnly : bool
(* Defaults:
    let_catalogOnly = defaultArg catalogOnly false
*)
-> unit
override DeleteLocation :
    location : DatabaseLocation *
    ?catalogOnly : bool
(* Defaults:
    let_catalogOnly = defaultArg catalogOnly false
*)
-> unit
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The DatabaseLocation to delete

catalogOnly (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Of true, only delete catalog entry. Leave the directory as is.

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.FlushUpdates Method

Send all updated pages to the server(s) managing those pages, freeing up memory in client.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushUpdates ()
```

VB

```
Public Overrides Sub FlushUpdates
```

C++

```
public:  
virtual void FlushUpdates () override
```

F#

```
abstract FlushUpdates : unit -> unit  
override FlushUpdates : unit -> unit
```

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.FlushUpdatesServers Method

Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushUpdatesServers ()
```

VB

```
Public Overrides Sub FlushUpdatesServers
```

C++

```
public:  
virtual void FlushUpdatesServers () override
```

F#

```
abstract FlushUpdatesServers : unit -> unit  
override FlushUpdatesServers : unit -> unit
```

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.InUseNumberOfPages Method

Gets the number of pages currently in use by a Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort InUseNumberOfPages (  
    Database db  
)
```

VB

```
Public Overrides Function InUseNumberOfPages (  
    db As Database  
) As UShort
```

C++

```
public:  
virtual unsigned short InUseNumberOfPages (  
    Database^ db  
) override
```

F#

```
abstract InUseNumberOfPages :  
    db : Database -> uint16  
override InUseNumberOfPages :  
    db : Database -> uint16
```

Parameters

db

Type: [VelocityDb.Database](#)

The Database to request info about

Return Value

Type: [UInt16](#)

The number of pages currently in use

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.NewDatabase Method

Create a new Database with a given database number

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database NewDatabase(
    uint dbNum,
    uint megaBytesPresize = 0,
    string name = null,
    bool signalError = true
)
```

VB

```
Public Overrides Function NewDatabase (
    dbNum As UInteger,
    Optional megaBytesPresize As UInteger = 0,
    Optional name As String = Nothing,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ NewDatabase(
    unsigned int dbNum,
    unsigned int megaBytesPresize = 0,
    String^ name = nullptr,
    bool signalError = true
) override
```

F#

```
abstract NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
    let_megaBytesPresize = defaultArg megaBytesPresize 0
    let_name = defaultArg name null
    let_signalError = defaultArg signalError true
*)
-> Database
override NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
```

VelocityDB Class Library

```
        let_megaBytesPresize = defaultArg megaBytesPresize 0
        let_name = defaultArg name null
        let_signalError = defaultArg signalError true
    *)
    -> Database
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the database to create

megaBytesPresize (Optional)

Type: [System.UInt32](#)

If you know that the Database will be large, presizing it may avoid file fragmentation. Default value is 0

name (Optional)

Type: [System.String](#)

Optionally name the new [Database](#)

signalError (Optional)

Type: [System.Boolean](#)

Optionally signal an error if creation of new [Database](#) fails

Return Value

Type: [Database](#)

The newly created [Database](#) or throws an exception if the [Database](#) already exist

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.NewLocation Method

Creates a new DatabaseLocation or updates existing ones

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override DatabaseLocation NewLocation(  
    DatabaseLocation location  
)
```

VB

```
Public Overrides Function NewLocation (  
    location As DatabaseLocation  
) As DatabaseLocation
```

C++

```
public:  
virtual DatabaseLocation^ NewLocation(  
    DatabaseLocation^ location  
) override
```

F#

```
abstract NewLocation :  
    location : DatabaseLocation -> DatabaseLocation  
override NewLocation :  
    location : DatabaseLocation -> DatabaseLocation
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The input location

Return Value

Type: [DatabaseLocation](#)

The new DatabaseLocation or an existing one

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.OpenAllDatabases Method

Open all databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
) override
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

Open for update?

Return Value

Type: [List\(Database\)](#)

[Missing <returns> documentation for

"M:VelocityDb.Session.ServerClientSession.OpenAllDatabases(System.Boolean)"]

VelocityDB Class Library

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.OpenDatabase Method

Opens a Database

Namespace: [VelocityDb.Session](#)**Assembly:** VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database OpenDatabase(
    uint dbNum,
    bool update = false,
    bool signalError = true
)
```

VB

```
Public Overrides Function OpenDatabase (
    dbNum As UInteger,
    Optional update As Boolean = false,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ OpenDatabase(
    unsigned int dbNum,
    bool update = false,
    bool signalError = true
) override
```

F#

```
abstract OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
override OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
```


VelocityDB Class Library

Parameters

dbNum

Type: [System.UInt32](#)

The Database number of the database to open

update (Optional)

Type: [System.Boolean](#)

Open the Database for update?

signalError (Optional)

Type: [System.Boolean](#)

Signal errors if errors found when opening the Database?

Return Value

Type: [Database](#)

The opened Database

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.OpenLocationDatabases Method

Opens all the databases in a given location

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenLocationDatabases (
    DatabaseLocation location,
    bool update = false
)
```

VB

```
Public Overrides Function OpenLocationDatabases (
    location As DatabaseLocation,
    Optional update As Boolean = false
) As List(Of Database)
```

C++

```
public:
virtual List<Database^>^ OpenLocationDatabases (
    DatabaseLocation^ location,
    bool update = false
) override
```

F#

```
abstract OpenLocationDatabases :
    location : DatabaseLocation *
    ?update : bool
(* Defaults:
    let_update = defaultArg update false
*)
-> List<Database>
override OpenLocationDatabases :
    location : DatabaseLocation *
    ?update : bool
(* Defaults:
    let_update = defaultArg update false
*)
-> List<Database>
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The location for which to open databases

update (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Shall each database be opened for update?

Return Value

Type: [List\(Database\)](#)

A list of opened databases

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.RestoreFrom Method

Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RestoreFrom(  
    DatabaseLocation backupLocation,  
    DateTime upToTime  
)
```

VB

```
Public Overrides Sub RestoreFrom (  
    backupLocation As DatabaseLocation,  
    upToTime As DateTime  
)
```

C++

```
public:  
virtual void RestoreFrom(  
    DatabaseLocation^ backupLocation,  
    DateTime upToTime  
) override
```

F#

```
abstract RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit  
override RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit
```

Parameters

backupLocation

Type: [VelocityDb.DatabaseLocation](#)

The location to restore from

upToTime

Type: [System.DateTime](#)

Restore location up to a given DateTime. Data backed up after this time will not be restored.

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.SubscribeToChanges Method

Subscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SubscribeToChanges (  
    Type aType,  
    string notifyIfTrueProperty = null  
)
```

VB

```
Public Overrides Sub SubscribeToChanges (  
    aType As Type,  
    Optional notifyIfTrueProperty As String = Nothing  
)
```

C++

```
public:  
virtual void SubscribeToChanges (  
    Type^ aType,  
    String^ notifyIfTrueProperty = nullptr  
) override
```

F#

```
abstract SubscribeToChanges :  
    aType : Type *  
    ?notifyIfTrueProperty : string  
(* Defaults:  
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null  
)  
-> unit  
override SubscribeToChanges :  
    aType : Type *  
    ?notifyIfTrueProperty : string  
(* Defaults:  
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null  
)  
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are interested in being notified about when persistent instances changes within a database.

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a boolean property. When property returns true, a change notification is send otherwise no notification is send. Leave as null to get a notification any time any instance of the selected type changes

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.UnsubscribeToChanges Method

Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void UnsubscribeToChanges (
    Type aType,
    string notifyIfTrueProperty = null
)
```

VB

```
Public Overrides Sub UnsubscribeToChanges (
    aType As Type,
    Optional notifyIfTrueProperty As String = Nothing
)
```

C++

```
public:
virtual void UnsubscribeToChanges (
    Type^ aType,
    String^ notifyIfTrueProperty = nullptr
) override
```

F#

```
abstract UnsubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
override UnsubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are no longer interested in being notified about when persistent instances changes within a database.

VelocityDB Class Library

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a Boolean property.

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSession.UpdateDatabase Method

Request an update lock on a database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateDatabase(  
    Database db  
)
```

VB

```
Public Overrides Function UpdateDatabase (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool UpdateDatabase(  
    Database^ db  
) override
```

F#

```
abstract UpdateDatabase :  
    db : Database -> bool  
override UpdateDatabase :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to update

Return Value

Type: [Boolean](#)

true if Database was updated, otherwise throws an exception

See Also

[ServerClientSession Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared Class

Use as a shared session for multiple reader threads connected to one or more VelocityDBServer(s).

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

[VelocityDb.Session.ServerClientSession](#)

VelocityDb.Session.ServerClientSessionShared

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ServerClientSessionShared : ServerClientSession
```

VB

```
Public Class ServerClientSessionShared
    Inherits ServerClientSession
```

C++


```
public ref class ServerClientSessionShared : public ServerClientSession
```

F#



```
type ServerClientSessionShared =
    class
        inherit ServerClientSession
    end
```

The **ServerClientSessionShared** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|------------------------|
|  | ServerClientSessionShared | Creates a new session. |












Properties














| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. (Overrides ServerClientSession.AddToIndexInSeperateThread.) |
|  | Databases | Gets a list of the currently opened databases (Overrides SessionBase.Databases.) |


| | | |
|--|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. (Overrides ServerClientSession.WriteToDiskInSeperateDatabaseThreads .) |
|--|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Methods










| | Name | Description |
|--|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Abort | Aborts a transaction, it undoes all persistent changes made within the transaction (Overrides ServerClientSession.Abort() .) |
| | BeginRead | Transaction control, begin a read only transaction (Overrides ServerClientSession.BeginRead(Boolean) .) |
| | BeginUpdate | Transaction control, begin an update transaction. (Overrides ServerClientSession.BeginUpdate(Boolean) .) |
| | Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() (Overrides SessionBase.Checkpoint() .) |
| | ClearCache | Clears page cache and closes databases. Avoid using this one for now. (Overrides SessionBase.ClearCache() .) |
| | ClearCachedPages | Clear cache of cached pages (Overrides SessionBase.ClearCachedPages() .) |
| | ClearPageCache | Clears cached pages from cache including page weak references. (Overrides SessionBase.ClearPageCache() .) |
| | Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides ServerClientSession.Compact() .) |
| | Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides ServerClientSession.Compact(Database) .) |
| | CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides ServerClientSession.CopyAllDatabasesTo(String, Boolean) .) |
| | CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is |

| | | |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | reopened. (Overrides SessionBase.CrossTransactionCache(Database, Boolean).) |
|  | DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides ServerClientSession.DeleteLocation(DatabaseLocation, Boolean).) |
|  | FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page will also be written. (Overrides SessionBase.FlushPageOf(OptimizedPersistable).) |
|  | FlushUpdates | Send all updated pages to the server(s) managing those pages, freeing up memory in client. (Overrides ServerClientSession.FlushUpdates().) |
|  | FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) (Overrides ServerClientSession.FlushUpdatesServers().) |
|  | ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. (Overrides SessionBase.ForceDatabaseCacheValidation().) |
|  | GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object (Overrides SessionBase.GlobalObjWrapperGet(Object, IOptimizedPersistable).) |
|  | NewDatabase | Create a new Database with a given database number (Overrides ServerClientSession.NewDatabase(UInt32, UInt32, String, Boolean).) |
|  | NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides ServerClientSession.NewLocation(DatabaseLocation).) |
|  | Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
|  | Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |
|  | Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  | Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OpenAllDatabases | Open all databases (Overrides ServerClientSession.OpenAllDatabases(Boolean).) |
|  OpenDatabase | Opens a Database (Overrides ServerClientSession.OpenDatabase(UInt32, Boolean, Boolean).) |
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides ServerClientSession.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  OpenSchema | Get the session active schema (Overrides SessionBase.OpenSchema(Boolean).) |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)).) |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)).) |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. (Overrides SessionBase.PossiblyFlushUpdatedPages(UInt32).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides ServerClientSession.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides ServerClientSession.SubscribeToChanges(Type, String).) |
|  Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable (Overrides SessionBase.Unpersist(Object).) |
|  UpdateDatabase | Request an update lock on a database (Overrides ServerClientSession.UpdateDatabase(Database).) |
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |

| | | |
|-----------------------------------------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. (Overrides SessionBase.Verify() .) |
|-----------------------------------------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv .) |
|  | ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport .) |
|  | ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport .) |
|  | ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport .) |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv .) |
|  | ImportJson(T) | (Defined by JsonImportExport .) |
|  | MicrosoftSync | (Defined by Sync .) |
|  | SyncWith(SessionBase) | Overloaded. (Defined by Sync .) |
|  | SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync .) |

See Also

[VelocityDb.Session Namespace](#)

ServerClientSessionShared Constructor

Creates a new session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ServerClientSessionShared(
    string systemDir,
    string systemHost = null,
    int waitForLockMilliseconds = 2000,
    bool optimisticLocking = true,
    bool inMemoryOnly = false,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes,
    bool enableDatabaseCache = true
)
```

VB

```
Public Sub New (
    systemDir As String,
    Optional systemHost As String = Nothing,
    Optional waitForLockMilliseconds As Integer = 2000,
    Optional optimisticLocking As Boolean = true,
    Optional inMemoryOnly As Boolean = false,
    Optional enablePageCache As Boolean = true,
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes,
    Optional enableDatabaseCache As Boolean = true
)
```

C++

```
public:
ServerClientSessionShared(
    String^ systemDir,
    String^ systemHost = nullptr,
    int waitForLockMilliseconds = 2000,
    bool optimisticLocking = true,
    bool inMemoryOnly = false,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes,
    bool enableDatabaseCache = true
)
```

F#

```
new :
    systemDir : string *
    ?systemHost : string *
    ?waitForLockMilliseconds : int *
    ?optimisticLocking : bool *
```



```
?inMemoryOnly : bool *
?enablePageCache : bool *
?objectCachingDefaultPolicy : CacheEnum *
?enableDatabaseCache : bool
(* Defaults:
    let_systemHost = defaultArg systemHost null
    let_waitForLockMilliseconds = defaultArg waitForLockMilliseconds 2000
    let_optimisticLocking = defaultArg optimisticLocking true
    let_inMemoryOnly = defaultArg inMemoryOnly false
    let_enablePageCache = defaultArg enablePageCache true
    let_objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
    let_enableDatabaseCache = defaultArg enableDatabaseCache true
*)
-> ServerClientSessionShared
```

Parameters

systemDir

Type: [System.String](#)

The startup location directory path (do not use UNC path). Use path as it is on the host owning the directory. Path can be an absolute (full) path or a path relative to **s_baseDatabasePath**. Change default of **s_baseDatabasePath** on server by starting server with a forth parameter specifying the base path

systemHost (Optional)

Type: [System.String](#)

Hostname of the host owning the systemDir

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

The desired maximum lock wait time

optimisticLocking (Optional)

Type: [System.Boolean](#)

Use optimistic locking. With optimistic locking, readers are always permitted and multiple updaters are permitted to update the same Databases/Pages but only the first transaction to commit or flush an updated Database/Page will be successful in making the commit change. Other updaters will get an `OptimisticLockingException` if they try to commit/flush updates to the same Pages/Database. See http://en.wikipedia.org/wiki/Optimistic_concurrency_control for further explanation of optimistic locking. If optimistic locking is not enabled then pessimistic locking is used. See: http://en.wikipedia.org/wiki/Concurrency_control

inMemoryOnly (Optional)

Type: [System.Boolean](#)

Disable all disk access by using this option. Best if used in combination with backup [DatabaseLocation](#). See High availability sample application

enablePageCache (Optional)

Type: [System.Boolean](#)

If strong reference page caching isn't desired, turn it of by setting to false

VelocityDB Class Library

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

Determines default value of [Cache](#)

enableDatabaseCache (Optional)

Type: [System.Boolean](#)

Is this session going to enable strong reference [Database](#) caching?

See Also




[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.ServerClientSessionShared Properties

The [ServerClientSessionShared](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. (Overrides ServerClientSession.AddToIndexInSeperateThread.) |
|  | Databases | Gets a list of the currently opened databases (Overrides SessionBase.Databases.) |
|  | WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. (Overrides ServerClientSession.WriteToDiskInSeperateDatabaseThreads.) |

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.AddToIndexInSeperateThread Property

Allow adding objects to indices to be done in a worker thread instead of in main thread.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AddToIndexInSeperateThread { get; set; }
```

VB

```
Public Overrides Property AddToIndexInSeperateThread As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool AddToIndexInSeperateThread {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract AddToIndexInSeperateThread : bool with get, set  
override AddToIndexInSeperateThread : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Databases Property

Gets a list of the currently opened databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Database> Databases { get; }
```

VB

```
Public Overrides ReadOnly Property Databases As IList(Of Database)  
    Get
```

C++

```
public:  
virtual property IList<Database^>^ Databases {  
    IList<Database^>^ get () override;  
}
```

F#

```
abstract Databases : IList<Database> with get  
override Databases : IList<Database> with get
```

Property Value

Type: [IList\(Database\)](#)

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.WriteToDiskInSeperateDatabaseThreads Property

Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. [AddToIndexInSeperateThread](#) must also be enabled for this to work.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool WriteToDiskInSeperateDatabaseThreads { get; set; }
```

VB

```
Public Overrides Property WriteToDiskInSeperateDatabaseThreads As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool WriteToDiskInSeperateDatabaseThreads {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract WriteToDiskInSeperateDatabaseThreads : bool with get, set  
override WriteToDiskInSeperateDatabaseThreads : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also













[ServerClientSessionShared Class](#)





[VelocityDb.Session Namespace](#)













ServerClientSessionShared.ServerClientSessionShared Methods

The [ServerClientSessionShared](#) type exposes the following members.










Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Aborts a transaction, it undoes all persistent changes made within the transaction (Overrides ServerClientSession.Abort() .) |
|  BeginRead | Transaction control, begin a read only transaction (Overrides ServerClientSession.BeginRead(Boolean) .) |
|  BeginUpdate | Transaction control, begin an update transaction. (Overrides ServerClientSession.BeginUpdate(Boolean) .) |
|  Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() (Overrides SessionBase.Checkpoint() .) |
|  ClearCache | Clears page cache and closes databases. Avoid using this one for now. (Overrides SessionBase.ClearCache() .) |
|  ClearCachedPages | Clear cache of cached pages (Overrides SessionBase.ClearCachedPages() .) |
|  ClearPageCache | Clears cached pages from cache including page weak references. (Overrides SessionBase.ClearPageCache() .) |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides ServerClientSession.Compact() .) |
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides ServerClientSession.Compact(Database) .) |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides ServerClientSession.CopyAllDatabasesTo(String, Boolean) .) |
|  CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened. (Overrides SessionBase.CrossTransactionCache(Database, Boolean) .) |
|  DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides |

| | | |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | ServerClientSession.DeleteLocation(DatabaseLocation, Boolean).) |
|  | FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written. (Overrides SessionBase.FlushPageOf(OptimizedPersistable).) |
|  | FlushUpdates | Send all updated pages to the server(s) managing those pages, freeing up memory in client. (Overrides ServerClientSession.FlushUpdates().) |
|  | FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) (Overrides ServerClientSession.FlushUpdatesServers().) |
|  | ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. (Overrides SessionBase.ForceDatabaseCacheValidation().) |
|  | GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object (Overrides SessionBase.GlobalObjWrapperGet(Object, IOptimizedPersistable).) |
|  | NewDatabase | Create a new Database with a given database number (Overrides ServerClientSession.NewDatabase(UInt32, UInt32, String, Boolean).) |
|  | NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides ServerClientSession.NewLocation(DatabaseLocation).) |
|  | Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
|  | Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |
|  | Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  | Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |
|  | OpenAllDatabases | Open all databases (Overrides ServerClientSession.OpenAllDatabases(Boolean).) |
|  | OpenDatabase | Opens a Database (Overrides ServerClientSession.OpenDatabase(UInt32, Boolean, Boolean).) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides ServerClientSession.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  OpenSchema | Get the session active schema (Overrides SessionBase.OpenSchema(Boolean).) |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)).) |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)).) |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. (Overrides SessionBase.PossiblyFlushUpdatedPages(UInt32).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides ServerClientSession.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides ServerClientSession.SubscribeToChanges(Type, String).) |
|  Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable (Overrides SessionBase.Unpersist(Object).) |
|  UpdateDatabase | Request an update lock on a database (Overrides ServerClientSession.UpdateDatabase(Database).) |
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |
|  Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. (Overrides SessionBase.Verify().) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  | ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  | ImportJson(T) | (Defined by JsonImportExport.) |
|  | MicrosoftSync | (Defined by Sync.) |
|  | SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  | SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[ServerClientSessionShared Class](#)[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Abort Method

Aborts a transaction, it undoes all persistent changes made within the transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Abort ()
```

VB

```
Public Overrides Sub Abort
```

C++

```
public:  
virtual void Abort () override
```

F#

```
abstract Abort : unit -> unit  
override Abort : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.BeginRead Method

Transaction control, begin a read only transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginRead(  
    bool doRecoveryCheck = false  
)
```

VB

```
Public Overrides Function BeginRead (  
    Optional doRecoveryCheck As Boolean = false  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginRead(  
    bool doRecoveryCheck = false  
) override
```

F#

```
abstract BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction  
override BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

By default we do not check for a failed update transaction that requires reverting one or more database changes. We don't do it by default because it can be a little costly. Change this parameter to `true` if you prefer ensured consistency over better performance or make change only if you encounter an exception in the transaction.

VelocityDB Class Library

Return Value

Type: [SessionBase.Transaction](#)

A transaction helper object.

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.BeginUpdate Method

Transaction control, begin an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginUpdate(  
    bool doRecoveryCheck  
)
```

VB

```
Public Overrides Function BeginUpdate (  
    doRecoveryCheck As Boolean  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginUpdate (  
    bool doRecoveryCheck  
) override
```

F#

```
abstract BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction  
override BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction
```

Parameters

doRecoveryCheck

Type: [System.Boolean](#)

Set to false only when moving system databases to a new directory

Return Value

Type: [SessionBase.Transaction](#)

A [SessionBase.Transaction](#) helper object.

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Checkpoint Method

Same as [Commit\(Boolean, Boolean\)](#) followed by [BeginUpdate\(\)](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction Checkpoint ()
```

VB

```
Public Overrides Function Checkpoint As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ Checkpoint () override
```

F#

```
abstract Checkpoint : unit -> SessionBase.Transaction  
override Checkpoint : unit -> SessionBase.Transaction
```

Return Value

Type: [SessionBase.Transaction](#)

[SessionBase.Transaction](#) started

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.ClearCache Method

Clears page cache and closes databases. Avoid using this one for now.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearCache ()
```

VB

```
Public Overrides Sub ClearCache
```

C++

```
public:  
virtual void ClearCache () override
```

F#

```
abstract ClearCache : unit -> unit  
override ClearCache : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.ClearCachedPages Method

Clear cache of cached pages

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearCachedPages ()
```

VB

```
Public Overrides Sub ClearCachedPages
```

C++

```
public:  
virtual void ClearCachedPages () override
```

F#

```
abstract ClearCachedPages : unit -> unit  
override ClearCachedPages : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.ClearPageCache Method

Clears cached pages from cache including page weak references.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearPageCache ()
```

VB

```
Public Overrides Sub ClearPageCache
```

C++

```
public:  
virtual void ClearPageCache () override
```

F#

```
abstract ClearPageCache : unit -> unit  
override ClearPageCache : unit -> unit
```



See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Compact Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides ServerClientSession.Compact() .) |
|  | Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides ServerClientSession.Compact(Database) .) |

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Compact Method

Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each [Database](#) file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact ()
```

VB

```
Public Overrides Sub Compact
```

C++

```
public:  
virtual void Compact () override
```

F#

```
abstract Compact : unit -> unit  
override Compact : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Compact Method (Database)

Reduce size of database, if possible, by truncating file where unused space begins

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact (  
    Database db  
)
```

VB

```
Public Overrides Sub Compact (  
    db As Database  
)
```

C++

```
public:  
virtual void Compact (  
    Database^ db  
) override
```

F#

```
abstract Compact :  
    db : Database -> unit  
override Compact :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

to compact

See Also

[ServerClientSessionShared Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.CopyAllDatabasesTo Method

Copies all databases to a selected directory on the local host.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void CopyAllDatabasesTo(  
    string directory,  
    bool systemDatabaseLocationOnly  
)
```

VB

```
Public Overrides Sub CopyAllDatabasesTo (  
    directory As String,  
    systemDatabaseLocationOnly As Boolean  
)
```

C++

```
public:  
virtual void CopyAllDatabasesTo(  
    String^ directory,  
    bool systemDatabaseLocationOnly  
) override
```

F#

```
abstract CopyAllDatabasesTo :  
    directory : string *  
    systemDatabaseLocationOnly : bool -> unit  
override CopyAllDatabasesTo :  
    directory : string *  
    systemDatabaseLocationOnly : bool -> unit
```

Parameters

directory

Type: [System.String](#)

Path to a directory

systemDatabaseLocationOnly

Type: [System.Boolean](#)

Only copy databases in system database location

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.CrossTransactionCache Method

By default databases are only referenced by a [WeakReference](#) across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void CrossTransactionCache (
    Database db,
    bool enable = true
)
```

VB

```
Public Overrides Sub CrossTransactionCache (
    db As Database,
    Optional enable As Boolean = true
)
```

C++

```
public:
virtual void CrossTransactionCache (
    Database^ db,
    bool enable = true
) override
```

F#

```
abstract CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
override CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
```

Parameters

db

VelocityDB Class Library

Type: [VelocityDb.Database](#)

The Database to cache

enable (Optional)

Type: [System.Boolean](#)

Add or remove strong reference. If true, add a strong reference

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.DeleteLocation Method

Deletes a [DatabaseLocation](#), location must first not have any Databases in it

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void DeleteLocation(  
    DatabaseLocation location,  
    bool catalogOnly = false  
)
```

VB

```
Public Overrides Sub DeleteLocation (  
    location As DatabaseLocation,  
    Optional catalogOnly As Boolean = false  
)
```

C++

```
public:  
virtual void DeleteLocation(  
    DatabaseLocation^ location,  
    bool catalogOnly = false  
) override
```

F#

```
abstract DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
)  
-> unit  
override DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
)  
-> unit
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The DatabaseLocation to delete

catalogOnly (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Of true, only delete catalog entry. Leave the directory as is.

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.FlushPageOf Method

By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushPageOf(  
    OptimizedPersistable pObj  
)
```

VB

```
Public Overrides Sub FlushPageOf (  
    pObj As OptimizedPersistable  
)
```

C++

```
public:  
virtual void FlushPageOf(  
    OptimizedPersistable^ pObj  
) override
```

F#

```
abstract FlushPageOf :  
    pObj : OptimizedPersistable -> unit  
override FlushPageOf :  
    pObj : OptimizedPersistable -> unit
```

Parameters

pObj

Type: [VelocityDb.OptimizedPersistable](#)

An object indicating what [Page](#) to flush

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.FlushUpdates Method

Send all updated pages to the server(s) managing those pages, freeing up memory in client.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushUpdates ()
```

VB

```
Public Overrides Sub FlushUpdates
```

C++

```
public:  
virtual void FlushUpdates () override
```

F#

```
abstract FlushUpdates : unit -> unit  
override FlushUpdates : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.FlushUpdatesServers Method

Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushUpdatesServers ()
```

VB

```
Public Overrides Sub FlushUpdatesServers
```

C++

```
public:  
virtual void FlushUpdatesServers () override
```

F#

```
abstract FlushUpdatesServers : unit -> unit  
override FlushUpdatesServers : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.ForceDatabaseCacheValidation Method

Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ForceDatabaseCacheValidation()
```

VB

```
Public Overrides Sub ForceDatabaseCacheValidation
```

C++

```
public:  
virtual void ForceDatabaseCacheValidation() override
```

F#

```
abstract ForceDatabaseCacheValidation : unit -> unit  
override ForceDatabaseCacheValidation : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.GlobalObjWrapperGet Method

Lookup wrapper object for a non IOptimizedPersistable object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool GlobalObjWrapperGet (
    Object lookupObj,
    out IOptimizedPersistable pObj
)
```

VB

```
Public Overrides Function GlobalObjWrapperGet (
    lookupObj As Object,
    <OutAttribute> ByRef pObj As IOptimizedPersistable
) As Boolean
```

C++

```
public:
virtual bool GlobalObjWrapperGet (
    Object^ lookupObj,
    [OutAttribute] IOptimizedPersistable^% pObj
) override
```

F#

```
abstract GlobalObjWrapperGet :
    lookupObj : Object *
    pObj : IOptimizedPersistable byref -> bool
override GlobalObjWrapperGet :
    lookupObj : Object *
    pObj : IOptimizedPersistable byref -> bool
```

Parameters

lookupObj

Type: [System.Object](#)

object to look for

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

wrapper object

Return Value

Type: [Boolean](#)

Wrapper object

VelocityDB Class Library

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.NewDatabase Method

Create a new Database with a given database number

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database NewDatabase(
    uint dbNum,
    uint megaBytesPresize = 0,
    string name = null,
    bool signalError = true
)
```

VB

```
Public Overrides Function NewDatabase (
    dbNum As UInteger,
    Optional megaBytesPresize As UInteger = 0,
    Optional name As String = Nothing,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ NewDatabase(
    unsigned int dbNum,
    unsigned int megaBytesPresize = 0,
    String^ name = nullptr,
    bool signalError = true
) override
```

F#

```
abstract NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
    let_megaBytesPresize = defaultArg megaBytesPresize 0
    let_name = defaultArg name null
    let_signalError = defaultArg signalError true
*)
-> Database
override NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
```

VelocityDB Class Library

```
        let_megaBytesPresize = defaultArg megaBytesPresize 0
        let_name = defaultArg name null
        let_signalError = defaultArg signalError true
    *)
    -> Database
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the database to create

megaBytesPresize (Optional)

Type: [System.UInt32](#)

If you know that the Database will be large, presizing it may avoid file fragmentation. Default value is 0

name (Optional)

Type: [System.String](#)

Optionally name the new [Database](#)

signalError (Optional)

Type: [System.Boolean](#)

Optionally signal an error if creation of new [Database](#) fails

Return Value

Type: [Database](#)

The newly created [Database](#) or throws an exception if the [Database](#) already exist

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.NewLocation Method

Creates a new DatabaseLocation or updates existing ones

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override DatabaseLocation NewLocation(  
    DatabaseLocation location  
)
```

VB

```
Public Overrides Function NewLocation (  
    location As DatabaseLocation  
) As DatabaseLocation
```

C++

```
public:  
virtual DatabaseLocation^ NewLocation(  
    DatabaseLocation^ location  
) override
```

F#

```
abstract NewLocation :  
    location : DatabaseLocation -> DatabaseLocation  
override NewLocation :  
    location : DatabaseLocation -> DatabaseLocation
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The input location

Return Value

Type: [DatabaseLocation](#)

The new DatabaseLocation or an existing one





See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Open Method

Overload List

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
|  Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |

See Also

[ServerClientSessionShared Class](#)[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Open Method (UInt64, Boolean, Boolean, Int32)

Opens a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Open(  
    ulong oid,  
    bool update,  
    bool inFlush,  
    int graphDepthToLoad = 2147483647  
)
```

VB

```
Public Overrides Function Open (  
    oid As ULong,  
    update As Boolean,  
    inFlush As Boolean,  
    Optional graphDepthToLoad As Integer = 2147483647  
) As Object
```

C++

```
public:  
virtual Object^ Open(  
    unsigned long long oid,  
    bool update,  
    bool inFlush,  
    int graphDepthToLoad = 2147483647  
) override
```

F#

```
abstract Open :  
    oid : uint64 *  
    update : bool *  
    inFlush : bool *  
    ?graphDepthToLoad : int  
(* Defaults:  
    let _graphDepthToLoad = defaultArg graphDepthToLoad 2147483647  
*)  
-> Object  
override Open :  
    oid : uint64 *  
    update : bool *  
    inFlush : bool *  
    ?graphDepthToLoad : int  
(* Defaults:
```

```
        let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> Object
```

Parameters

oid

Type: [System.UInt64](#)

The Id of the object to open

update

Type: [System.Boolean](#)

Open the object for update?

inFlush

Type: [System.Boolean](#)

if true, disallow page flushing while opening the object

graphDepthToLoad (Optional)

Type: [System.Int32](#)

Set this if you want to limit the depth of the graph loaded by this open.

Return Value

Type: [Object](#)

A persistent object

See Also

[ServerClientSessionShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Open Method (UInt32, UInt16, UInt16, Boolean, Int32)

Opens a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(  
    uint dbNum,  
    ushort pageNum,  
    ushort slotNum,  
    bool update,  
    int graphDepthToLoad = 2147483647  
)
```

VB

```
Public Overrides Function Open (  
    dbNum As UInteger,  
    pageNum As UShort,  
    slotNum As UShort,  
    update As Boolean,  
    Optional graphDepthToLoad As Integer = 2147483647  
) As IOptimizedPersistable
```

C++

```
public:  
virtual IOptimizedPersistable^ Open(  
    unsigned int dbNum,  
    unsigned short pageNum,  
    unsigned short slotNum,  
    bool update,  
    int graphDepthToLoad = 2147483647  
) override
```

F#

```
abstract Open :  
    dbNum : uint32 *  
    pageNum : uint16 *  
    slotNum : uint16 *  
    update : bool *  
    ?graphDepthToLoad : int  
(* Defaults:  
    let graphDepthToLoad = defaultArg graphDepthToLoad 2147483647  
)  
-> IOptimizedPersistable  
override Open :  
    dbNum : uint32 *
```

VelocityDB Class Library

```
    pageNum : uint16 *
    slotNum : uint16 *
    update : bool *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the object to open

pageNum

Type: [System.UInt16](#)

Page number of the object to open

slotNum

Type: [System.UInt16](#)

Slot number of the object to open

update

Type: [System.Boolean](#)

Open the object for update?

graphDepthToLoad (Optional)

Type: [System.Int32](#)

Limit depth of graph to open

Return Value

Type: [IOptimizedPersistable](#)

The opened object or null if it does not exist

See Also

[ServerClientSessionShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Open Method (Database, UInt64, Boolean, List<IOptimizedPersistable>, Boolean, Int32, Int32)

For internal usage only, used in generated code

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(
    Database db,
    ulong oid,
    bool update,
    List<IOptimizedPersistable> toLoadMembers,
    bool inFlush,
    int graphDepth = 0,
    int graphDepthToLoad = 2147483647
)
```

VB

```
Public Overrides Function Open (
    db As Database,
    oid As ULong,
    update As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    inFlush As Boolean,
    Optional graphDepth As Integer = 0,
    Optional graphDepthToLoad As Integer = 2147483647
) As IOptimizedPersistable
```

C++

```
public:
virtual IOptimizedPersistable^ Open(
    Database^ db,
    unsigned long long oid,
    bool update,
    List<IOptimizedPersistable^>^ toLoadMembers,
    bool inFlush,
    int graphDepth = 0,
    int graphDepthToLoad = 2147483647
) override
```

F#

```
abstract Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
```

```
        ?graphDepth : int *
        ?graphDepthToLoad : int
(* Defaults:
    let_graphDepth = defaultArg graphDepth 0
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
override Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
    ?graphDepth : int *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepth = defaultArg graphDepth 0
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

oid

Type: [System.UInt64](#)

[Missing <param name="oid"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

update

Type: [System.Boolean](#)

[Missing <param name="update"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

inFlush

Type: [System.Boolean](#)

[Missing <param name="inFlush"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

graphDepth (Optional)

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

graphDepthToLoad (Optional)

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

Return Value

Type: [IOptimizedPersistable](#)

[Missing <returns> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

See Also

[ServerClientSessionShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Open Method (Database, UInt64, Boolean, List<IOptimizedPersistable>, Boolean, Int32, Int32, Boolean)

For internal use only in generated code

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(
    Database db,
    ulong oid,
    bool update,
    List<IOptimizedPersistable> toLoadMembers,
    bool inFlush,
    int graphDepth,
    int graphDepthToLoad,
    bool signalNotExistError = false
)
```

VB

```
Public Overrides Function Open (
    db As Database,
    oid As ULong,
    update As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    inFlush As Boolean,
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    Optional signalNotExistError As Boolean = false
) As IOptimizedPersistable
```

C++

```
public:
virtual IOptimizedPersistable^ Open(
    Database^ db,
    unsigned long long oid,
    bool update,
    List<IOptimizedPersistable^>^ toLoadMembers,
    bool inFlush,
    int graphDepth,
    int graphDepthToLoad,
    bool signalNotExistError = false
) override
```

F#

```
abstract Open :
    db : Database *
    oid : uint64 *
```

```
        update : bool *
        toLoadMembers : List<IOptimizedPersistable> *
        inFlush : bool *
        graphDepth : int *
        graphDepthToLoad : int *
        ?signalNotExistError : bool
(* Defaults:
    let_signalNotExistError = defaultArg signalNotExistError false
*)
-> IOptimizedPersistable
override Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
    graphDepth : int *
    graphDepthToLoad : int *
    ?signalNotExistError : bool
(* Defaults:
    let_signalNotExistError = defaultArg signalNotExistError false
*)
-> IOptimizedPersistable
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDb.Session.ServerClientSessionShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32,System.Boolean)"]

oid

Type: [System.UInt64](#)

Id of the object to open, must match with Database id

update

Type: [System.Boolean](#)

Update object when opening it

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

Internal list of objects to be loaded as part of graph load

inFlush

Type: [System.Boolean](#)

Permit page flushes as side affect of opening object?

graphDepth

Type: [System.Int32](#)

Current graph depth of graph load

VelocityDB Class Library

graphDepthToLoad

Type: [System.Int32](#)

Max graph depth to load

signalNotExistError (Optional)

Type: [System.Boolean](#)

Signal an exception if object/page/database does not exist.

Return Value

Type: [IOptimizedPersistable](#)

a persistent object

See Also

[ServerClientSessionShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.OpenAllDatabases Method

Open all databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
) override
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

Open for update?

Return Value

Type: [List\(Database\)](#)

[Missing <returns> documentation for

"M:VelocityDb.Session.ServerClientSessionShared.OpenAllDatabases(System.Boolean)"]

VelocityDB Class Library

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.OpenDatabase Method

Opens a Database

Namespace: [VelocityDb.Session](#)**Assembly:** VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database OpenDatabase(
    uint dbNum,
    bool update = false,
    bool signalError = true
)
```

VB

```
Public Overrides Function OpenDatabase (
    dbNum As UInteger,
    Optional update As Boolean = false,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ OpenDatabase(
    unsigned int dbNum,
    bool update = false,
    bool signalError = true
) override
```

F#

```
abstract OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
override OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
```

VelocityDB Class Library

Parameters

dbNum

Type: [System.UInt32](#)

The Database number of the database to open

update (Optional)

Type: [System.Boolean](#)

Open the Database for update?

signalError (Optional)

Type: [System.Boolean](#)

Signal errors if errors found when opening the Database?

Return Value

Type: [Database](#)

The opened Database

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.OpenLocationDatabases Method

Opens all the databases in a given location

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenLocationDatabases (  
    DatabaseLocation location,  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenLocationDatabases (  
    location As DatabaseLocation,  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenLocationDatabases (  
    DatabaseLocation^ location,  
    bool update = false  
) override
```

F#

```
abstract OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
)  
-> List<Database>  
override OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
)  
-> List<Database>
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The location for which to open databases

update (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Shall each database be opened for update?

Return Value

Type: [List\(Database\)](#)

A list of opened databases

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.OpenSchema Method

Get the session active schema

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override Schema OpenSchema (  
    bool update  
)
```

```
VB  
Public Overrides Function OpenSchema (  
    update As Boolean  
) As Schema
```

```
C++  
public:  
virtual Schema^ OpenSchema (  
    bool update  
) override
```

```
F#  
abstract OpenSchema :  
    update : bool -> Schema  
override OpenSchema :  
    update : bool -> Schema
```

Parameters

update

Type: [System.Boolean](#)

Do update the schema?

Return Value

Type: [Schema](#)

The active schema



See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Persist Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)) .) |
|  | Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) .) |

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Persist Method (IOptimizedPersistable, Nullable(UInt16))

This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides [PlacementDatabaseNumber](#) and returns something other than [DefaultPlacementDatabaseNumber](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    IOptimizedPersistable ipObj,
    ushort? objectsPerPage = null
)
```

VB

```
Public Overrides Function Persist (
    ipObj As IOptimizedPersistable,
    Optional objectsPerPage As UShort? = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    IOptimizedPersistable^ ipObj,
    Nullable<unsigned short> objectsPerPage = nullptr
) override
```

F#

```
abstract Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let _objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
override Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let _objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
```

Parameters

ipObj

VelocityDB Class Library

Type: [VelocityDb.IOptimizedPersistable](#)

The object to make persistent

objectsPerPage (Optional)

Type: [System.Nullable\(UInt16\)](#)

Override of objects per page, only respected when persisting first object of some type

Return Value

Type: [UInt64](#)

The Id of the persisted object

See Also

[ServerClientSessionShared Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Persist Method (Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable))

Persists an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    IOptimizedPersistable pObj,
    Schema schema,
    ushort slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    pObj As IOptimizedPersistable,
    schema As Schema,
    Optional slotLimitPerPage As UShort = 0,
    Optional inFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    IOptimizedPersistable^ pObj,
    Schema^ schema,
    unsigned short slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
```

```
(* Defaults:
    let_slotLimitPerPage = defaultArg slotLimitPerPage 0
    let_inFlush = defaultArg inFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_slotLimitPerPage = defaultArg slotLimitPerPage 0
    let_inFlush = defaultArg inFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

Placement object determining where to persist the object

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to persist

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active session schema

slotLimitPerPage (Optional)

Type: [System.UInt16](#)

Override of how many slots per page to permit

inFlush (Optional)

Type: [System.Boolean](#)

Set to true to disallow page flushes as a side affect

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A list of objects waiting to be persisted

Return Value

Type: [UInt64](#)

Id of persisted object

See Also

[ServerClientSessionShared Class](#)

VelocityDB Class Library

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.PossiblyFlushUpdatedPages Method

Call this function if you may have updated many pages to possibly free up memory if too much memory is in use.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void PossiblyFlushUpdatedPages (
    uint numberOfUpdatedPagesPermittedWithoutFlushCheck = 100
)
```

VB

```
Public Overrides Sub PossiblyFlushUpdatedPages (
    Optional numberOfUpdatedPagesPermittedWithoutFlushCheck As UInteger =
100
)
```

C++

```
public:
virtual void PossiblyFlushUpdatedPages (
    unsigned int numberOfUpdatedPagesPermittedWithoutFlushCheck = 100
) override
```

F#

```
abstract PossiblyFlushUpdatedPages :
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32
(* Defaults:
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg
numberOfUpdatedPagesPermittedWithoutFlushCheck 100
*)
-> unit
override PossiblyFlushUpdatedPages :
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32
(* Defaults:
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg
numberOfUpdatedPagesPermittedWithoutFlushCheck 100
*)
-> unit
```

Parameters

numberOfUpdatedPagesPermittedWithoutFlushCheck (Optional)

Type: [System.UInt32](#)

Only do check if the number of updated pages is greater than this number

See Also

[ServerClientSessionShared Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.RestoreFrom Method

Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RestoreFrom(  
    DatabaseLocation backupLocation,  
    DateTime upToTime  
)
```

VB

```
Public Overrides Sub RestoreFrom (  
    backupLocation As DatabaseLocation,  
    upToTime As DateTime  
)
```

C++

```
public:  
virtual void RestoreFrom(  
    DatabaseLocation^ backupLocation,  
    DateTime upToTime  
) override
```

F#

```
abstract RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit  
override RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit
```

Parameters

backupLocation

Type: [VelocityDb.DatabaseLocation](#)

The location to restore from

upToTime

Type: [System.DateTime](#)

Restore location up to a given DateTime. Data backed up after this time will not be restored.

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.SubscribeToChanges Method

Subscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SubscribeToChanges (
    Type aType,
    string notifyIfTrueProperty = null
)
```

VB

```
Public Overrides Sub SubscribeToChanges (
    aType As Type,
    Optional notifyIfTrueProperty As String = Nothing
)
```

C++

```
public:
virtual void SubscribeToChanges (
    Type^ aType,
    String^ notifyIfTrueProperty = nullptr
) override
```

F#

```
abstract SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
override SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are interested in being notified about when persistent instances changes within a database.

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a boolean property. When property returns true, a change notification is send otherwise no notification is send. Leave as null to get a notification any time any instance of the selected type changes

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Unpersist Method

Use this when you want to delete objects that are not assignable as [IOptimizedPersistable](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override bool Unpersist(  
    Object obj  
)
```

```
VB  
Public Overrides Function Unpersist (  
    obj As Object  
) As Boolean
```

```
C++  
public:  
virtual bool Unpersist(  
    Object^ obj  
) override
```

```
F#  
abstract Unpersist :  
    obj : Object -> bool  
override Unpersist :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to be deleted from persistent storage

Return Value

Type: [Boolean](#)

true if object found and deleted; otherwise false

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.UpdateDatabase Method

Request an update lock on a database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateDatabase(  
    Database db  
)
```

VB

```
Public Overrides Function UpdateDatabase (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool UpdateDatabase(  
    Database^ db  
) override
```

F#

```
abstract UpdateDatabase :  
    db : Database -> bool  
override UpdateDatabase :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to update

Return Value

Type: [Boolean](#)

true if Database was updated, otherwise throws an exception



See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.UpdateObject Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  | UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.UpdateObject Method (Object)

Tag an object as updated so that it will be updated persistently

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateObject(  
    Object obj  
)
```

VB

```
Public Overrides Function UpdateObject (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject (  
    Object^ obj  
) override
```

F#

```
abstract UpdateObject :  
    obj : Object -> bool  
override UpdateObject :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to update.

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

See Also

[ServerClientSessionShared Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.UpdateObject Method (IOptimizedPersistable, Boolean, Boolean)

Updates an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateObject(
    IOptimizedPersistable obj,
    bool inFlush,
    bool deleteObjFromIndexes = true
)
```

VB

```
Public Overrides Function UpdateObject (
    obj As IOptimizedPersistable,
    inFlush As Boolean,
    Optional deleteObjFromIndexes As Boolean = true
) As Boolean
```

C++

```
public:
virtual bool UpdateObject(
    IOptimizedPersistable^ obj,
    bool inFlush,
    bool deleteObjFromIndexes = true
) override
```

F#

```
abstract UpdateObject :
    obj : IOptimizedPersistable *
    inFlush : bool *
    ?deleteObjFromIndexes : bool
(* Defaults:
    let _deleteObjFromIndexes = defaultArg deleteObjFromIndexes true
*)
-> bool
override UpdateObject :
    obj : IOptimizedPersistable *
    inFlush : bool *
    ?deleteObjFromIndexes : bool
(* Defaults:
    let _deleteObjFromIndexes = defaultArg deleteObjFromIndexes true
*)
-> bool
```

VelocityDB Class Library

Parameters

obj

Type: [VelocityDb.IOptimizedPersistable](#)

Object to be updated

inFlush

Type: [System.Boolean](#)

Are we doing this as part of a page flush? Set to true to avoid a recursive page flush

deleteObjFromIndexes (Optional)

Type: [System.Boolean](#)

Set to false if you know that this object is not part of any index. Safe to leave set to true in any case

Return Value

Type: [Boolean](#)

True if object is updated; otherwise false

See Also

[ServerClientSessionShared Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

ServerClientSessionShared.Verify Method

Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Verify()
```

VB

```
Public Overrides Sub Verify
```

C++

```
public:  
virtual void Verify() override
```

F#

```
abstract Verify : unit -> unit  
override Verify : unit -> unit
```

See Also

[ServerClientSessionShared Class](#)

[VelocityDb.Session Namespace](#)

SessionBase Class

Transaction control, database creation and special persistent object api. Each session can use 4,294,967,295 [Databases](#). Each [Database](#) may have up to 65,535 [Pages](#) and a page can be up to 2 GB in size (.Net restriction of a byte array). The maximum data size a single session can manage is: $4,294,967,295 * 65,535 * 2147483648 = 604453686294542391705600$ bytes or about half a Yottabyte <http://en.wikipedia.org/wiki/Yottabyte>

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Session.SessionBase

[VelocityDb.Session.ServerClientSession](#)

[VelocityDb.Session.SessionNoServer](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class SessionBase : IEnumerable<Database>,
    IEnumerable, IDisposable, IEnlistmentNotification
```

VB

```
Public MustInherit Class SessionBase
    Implements IEnumerable(Of Database), IEnumerable,
    IDisposable, IEnlistmentNotification
```

C++














```
public ref class SessionBase abstract : IEnumerable<Database^>,
    IEnumerable, IDisposable, IEnlistmentNotification
```




















F#























```
[<AbstractClassAttribute>]
type SessionBase =
    class
        interface IEnumerable<Database>
        interface IEnumerable
        interface IDisposable
        interface IEnlistmentNotification
    end
```





The **SessionBase** type exposes the following members.

Properties
















| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. |
|  AssumeLocalHost | SessionNoServer ignores DatabaseLocation host value, it assumes localhost, ServerSessionClient does not assume local host, it follows the DatabaseLocation HostName value. |
|   BaseDatabasePath | If specified boot path (system database path) in session constructor isn't an absolute path then this base path is prefixed to make an absolute path. |
|   BTreeAddFastTransientBatchSize | Default batch size for BTree AddFast api, initially set to 100 |
|   ClearAllCachedObjectsWhenDetectingUpdatedDatabase | By default we clear all cached object when at the beginning of a transaction we detect an updated Database within our cached databases. Same logic is applied when calling ForceDatabaseCacheValidation() . We do so because most of the time an object in one database have (strong) references to objects in other databases. If Database A is NOT updated but database B is and an object in database A is cached with a strong reference to an object in database B then our cached object in database A may contain a stale/incorrect reference to object in database B. This is why we, by default, invalidate all cached objects when we detect a modified database within our cached databases. However, if you design your database schema without strong references between databases then you do not need to invalidate cached objects for other databases, only the ones that have changed. By consistently using WeakIOptimizedPersistableReference(T) , BTreeSet(Key) and BTreeMap(Key, Value) , you can keep references weak between databases. In such cases, set this property to false. |
|  ClientCache | Gets the client page cache object |
|  DatabaseLocations | Gets the DatabaseLocations DatabaseLocations for the session. |
|  Databases | Gets a list of the currently opened databases |
|   DefaultCompressPages | By default false, set to true if you want to change the default for newly created DatabaseLocation . |













| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DefaultNumberOfObjectsPerPage  | Default number of objects per page is 9999 |
|  DefaultStringComparer  | Set the StringComparer to use when no application provided comparator is provided. By default we will then use StringComparer.Ordinal. |
|  DeletedOids | Collected Oid for unpersisted objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  DoWindowsAuthentication  | By default do not use windows authentication when talking to VelocitydbServer's on other hosts. |
|  EmbedSerializableLists  | Starting in version 7.1.6, List(T) bits are NOT by default stored with its parent object, instead each such List(T) is stored with its own object Id. This change was necessary in order to handle circular references via List(T) objects within an ISerializable serialization. Set this value to true if you have existing persisted ISerializable with List(T) or else we will try to read the lists as non embedded when they are not. |
|  EnableAutoPageFlush | Pages may be flushed when memory available is low unless you set tis property to false. You would want it to be enabled unless you are trying to share a non SessionNoServerShared session by working on different databases in multiple threads. As in Wikipedia sample application. |
|  EnableSyncByTrackingChanges | Enable database and page change tracking. Turn off within an update transaction so that prior change set can be deleted. |
|  FileBytesChunkSize  | When transferring complete files from a server, we send the file bytes in chunks of this size. |
|  FileShareReader | FileShare for reader is ReadWrite with optimistic locking; otherwise Read |
|  FileShareUpdater | FileShare for updater is Read with optimistic locking; otherwise None |
|  InCommit | Indicates if we are within a transaction commit. |
|  InMemoryOnly | Gets information about this session, is it an in-memory only session or a session that persists data |
|  InTransaction | Indicates if we are within an active transaction. |
|  InUpdateTransaction | Indicates if we are within an active update transaction. |





















| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IsDisposed | Check if session has been disposed |
|  LocalHost  | Get the host name of your local host (the machine you are using) |
|  MaxStringLengthToIntern  | Get the maximum size a string must have before VelocityDB internals interns, Intern(String) , string when read from a Database . |
|  MaxUpdateTransactionLogHistory  | Determines max number of UpdateTransaction objects we maintain in update transaction log in Database 0, initially set to 9990 |
|  MinStringLengthToIntern  | Get the minimum size a string must have before VelocityDB internals interns, Intern(String) , string when read from a Database . |
|  NewDatabases | Get a list of all new databases created within this transaction |
|  NewOids | Collected Oid for newly persisted objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  NotifyBeforeCommit | Set to an Action of your choice if you want to get notified right before commit of an update transaction. |
|  ObjectCachingDefaultPolicy | Get/Set default value for Cache |
|  OptimisticLocking | Is the active Database and/or Page locking using optimistic locking model or not? See https://en.wikipedia.org/wiki/Optimistic_concurrency_control and https://en.wikipedia.org/wiki/Record_locking |
|  StrongReferenceDatabaseCaching | Is current session caching Databases using strong references or just by weak references. Set in session constructor or by calling CrossTransactionCacheAllDatabases(Boolean) . |
|  SystemDirectory | Get the path to the directory of the system Databases (0.oddb, 1.oddb, 2.oddb ...) of this session. |
|  SystemHostName | Get the host name of the system Databases (0.oddb, 1.oddb, 2.oddb ...) of this session. |
|  TextEncoding  | All text encoding is done with utf8 |
|  TraceDBActivity | |
|  TraceIndexUsage | If set to true, each time an index (BTreeSet) is used in a LINQ query expression, we output a line about that to the console output. |
|  TransactionNumber | Gets the transaction number of the current session |























| | |
|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  UpdatedOids | Collected Oid for updated objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  UseExternalStorageApi | This is specifically for Windows Phone API, set to true when you want to access databases from a memory card. |
|  WaitForLockMilliseconds | Get/Set the max time waiting for a lock to clear with pessimistic locking |
|  WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. |













Methods















| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Transaction control, abort the current transaction |
|  AllObjects(T) | Gets an object used for enumerating all objects in all Databases |
|  AnyPageWritten | Find out if any pages have been written for a given database in this update transaction |
|  BeginRead | Transaction control, begin a read only transaction |
|  BeginUpdate() | Transaction control, begin an update transaction. |
|  BeginUpdate(Boolean) | Transaction control, begin an update transaction. |
|  Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() |
|  ClearCache | Clears page cache and closes databases. Avoid using this one for now. |
|  ClearCachedObjects() | Clears the cache of cached objects. |
|  ClearCachedObjects(Database) | Clears the cache of cached objects within a Database . |
|  ClearCashedPages | Clear cache of cached pages |
|  ClearPageCache | Clears cached pages from cache including page weak references. |
|  CloseDatabase | Closes a Database |
|  Commit | Transaction control, commit the current transaction |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. |











| | |
|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins |
|  CompareTo | The default compare of two SessionBase is by system (bootup) host name and directory |
|  ContainsDatabase | Check if database file exist in a given DatabaseLocation |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. |
|  CreateDirectory | Creates a directory |
|  CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened. |
|  CrossTransactionCacheAllDatabases | By default databases are referenced by strong references instead of WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to all Databases so the cached Database may be used if version wasn't changed by a different thread since prior transaction. A strong reference is removed for any database that get invalidated due to a transaction abort or if another transaction commits a change to it. Check current session state with StrongReferenceDatabaseCaching |
|  DatabaseNumberOf | Get the database number associated with a specific type. Your application can store any type of object in any database but if the application exclusively use Persist(IOptimizedPersistable, Nullable(UInt16)) or Persist(Object) then this is the database Id that will be used given a type. |
|  DatabaseStillExist | Check if Database still exist |
|  DefaultDatabaseLocation | Gets the default (startup) DatabaseLocation |
|  DeleteDatabase | Delete a database, actual database file delete happens after a successful transaction commit. |
|  DeleteFile | Delete a file. For internal use. |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it |
|  DeleteObject | A way to delete an object without requiring that the object be opened first |
|  DeletePage | Deletes a page from a database. (part of database Sync beta version - not yet fully designed and tested) Used by SyncWith when syncing databases with another replica of the current databases |
|  DeployGenerateReaderWriter | Part of code generator, not yet ready for public release |
|  DeployInternalTypes | For internal use only |
|  Dispose | Closes this session |
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | Opens a Database file for read/update. Internal use only. |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | For internal use. |
|  FirstFreePage | Gets the page number of the first unallocated page in the specified Database |
|  FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written. |
|  FlushUpdates() | Write all updated data. This may free up some memory. |
|  FlushUpdates(Database) | Write any updated/new pages for a specific Database |
|  FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) |
|  ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. |
|  GetEnumerator | Enumerates all open databases for this session |
|  GetVersion | Gets the version of Database 0 |
|  GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object |
|  HighestInUseDatabaseNumber | Figure out what the highest in use database number is and return it. |
|  HighestInUseLocalDatabaseNumber | Figure out what the highest in use database number is on the local host and return it. |
|  IdOf | Finds the object id of a persistent object |













| | |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Index(T)() | Gets an index of all objects of a certain type |
|  Index(T)(String) | Gets an index of all objects of a certain type |
|  Index(T)(Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used). If you have updated objects persisted in a prior transaction that you want part of an index. Call FlushUpdates() or call Write()/> to add such an object to index before querying using an index./> |
|  Index(T)(String, Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used) |
|  InUseNumberOfPages | Gets the number of pages currently in use by a Database |
|  IsSameHost | Determine if two host names represent the same tcp/ip host |
|  IsWorkerThread | Tell if code is executed by index or page write worker thread |
|  LoadFields | Loads all field values of an object if they are not already loaded. |
|  LocateDb | Lookup the DatabaseLocation of a Database with a specified database number. |
|  NewDatabase | Create a new Database with a given database number |
|  NewLocation | Creates a new DatabaseLocation or updates existing ones |
|  Objects(T) | Enumerates all the objects of a given type |
|  OfType | Gets an object used for enumerating all objects in all Databases |
|  OidOf | Finds the object id of a persistent object |
|  OidShortOf | Finds the object id of a persistent object |
|  OpenAllDatabases | Open all databases |
|  OpenDatabase | Opens a Database |
|  OpenDatabaseLocations | Open the DatabaseLocations object and store it in the session. |
|  OpenLocationDatabases | Opens all the databases in a given location |
|  OpenPage | Opens a page for read, used by DatabaseManager for page browsing |
|  OpenSchema | Get the session active schema |
|  ParseCsvRow | Parses a row in a csv file and returns an iterator of column string values |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Persist(Object) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database. |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . |
|  Persist(Placement, IOptimizedPersistable) | Persist an object that is an OptimizedPersistable or a subclass |
|  Persist(Object, IOptimizedPersistable, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |
|  Persist(Object, Placement, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. |
|  RegisterClass | Register a class in the persistent schema. It is a good idea to do this explicitly instead of having it happen the first time an object of an unregistered class is made persistent |
|  RelocateDatabaseLocationFor | After moving/copying directory containing your Database with id dbId call this before starting a transaction to update host and directory of the associated DatabaseLocation . This function does not move the database files, it only updates the DatabaseLocation . |
|  RelocateDefaultDatabaseLocation | After moving/copying directory containing your Databases 0.oddb, 1.oddb ... call this before starting a transaction to update host and directory of the default DatabaseLocation . This function does not move the database files, it only updates the DatabaseLocation . |
|  ReplacePage | Only to be used in VelocityDbExtensions while syncing databases |
|  ReplacePersistedType(String, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old |










| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | Type loaded. Be sure to backup all your data before doing this. |
|  | ReplacePersistedType(Type, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old Type loaded. Be sure to backup all your data before doing this. |
|  | RestoreFrom(DatabaseLocation, DateTime) | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. |
|  | RestoreFrom(String, String, UInt32, DateTime) | Restores a DatabaseLocation |
|   | SetMinMaxStringIntern | A persistent string read into memory from a Database is interned using Intern(String) to speed up comparisons and conserve memory (when the same string occurs in many places) |
|  | SetTraceAllDbActivity | Enable some Trace output about persistent events related to all databases and pages. Add a Trace listener to get to Console window: Trace.Listeners.Add(new ConsoleTraceListener()); |
|  | SetTraceDbActivity | Enable some Trace output about persistent events related to a Database. Add a Trace listener to get to Console window: Trace.Listeners.Add(new ConsoleTraceListener()); |
|   | StringToType | Gets a Type given a string representation of a type |
|  | SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. |
|  | TryDatabase | Gets the current database to try |
|  | Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable |
|  | UnsubscribeToChanges | Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true. |
|  | UpdateClass | Let VelocityDB know that changes have been made to a class so that the schema manager will create a new TypeVersion and use the new version for all new objects of the specified Type. |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  UpdateDatabase | Request an update lock on a database |
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently |
|  UpdateObject(IOptimizedPersistable) | Tag an object as updated so that it will be updated persistently |
|  UpdateObject(Object, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
|  UpdateObject(IOptimizedPersistable, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object |
|  UpdatePage | Update a Page . Used by VelocityDBExtensions project |
|  Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. |
|  WaitForIndexUpdates | Internal Use and in VelocityDBExtensions |
|  WritePageBytes | |

Fields

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Assembly  | |
|  IndexDescriptorDatabaseNumber  | Database 3 is dedicated to keeping track of index usage |
|  s_serverTcpIpPortNumber  | By default port number 7031 (7032 if .net core) is used for communication with VelocityDBServer. Change the value here and also in VelocityDbServer.exe.config if you need to use a different port number. |
|  s_typeDatabaseIdOffsetFromTypeShortId  | Determines database Id to use for a certain VelocityDbType instance |
|  s_typeDatabaseIdOverflowIndexCollisionOffset  | If additional databases are required for storing instances of some index, start by attempting creating database at this offset from original |
|  s_typeDatabaseIdOverflowOffset  | If additional databases are required for storing instances of some type, start by attempting creating database at this offset from original |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  | ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  | ImportJson(T) | (Defined by JsonImportExport.) |
|  | MicrosoftSync | (Defined by Sync.) |
|  | SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  | SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |












See Also















[VelocityDb.Session Namespace](#)

















SessionBase.SessionBase Properties







The [SessionBase](#) type exposes the following members.

Properties

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. |
|  AssumeLocalHost | SessionNoServer ignores DatabaseLocation host value, it assumes localhost, ServerSessionClient does not assume local host, it follows the DatabaseLocation HostName value. |
|  BaseDatabasePath  | If specified boot path (system database path) in session constructor isn't an absolute path then this base path is prefixed to make an absolute path. |
|  BTreeAddFastTransientBatchSize  | Default batch size for BTree AddFast api, initially set to 100 |
|  ClearAllCachedObjectsWhenDetectingUpdatedDatabase  | By default we clear all cached object when at the beginning of a transaction we detect an updated Database within our cached databases. Same logic is applied when calling ForceDatabaseCacheValidation() . We do so because most of the time an object in one database have (strong) references to objects in other databases. If Database A is NOT updated but database B is and an object in database A is cached with a strong reference to an object in database B then our cached object in database A may contain a stale/incorrect reference to object in database B. This is why we, by default, invalidate all cached objects when we detect a modified database within our cached databases. However, if you design your database schema without strong references between databases then you do not need to invalidate cached objects for other databases, only the ones that have changed. By consistently using WeakIOptimizedPersistableReference(T) , BTreeSet(Key) and BTreeMap(Key, Value) , you can keep references weak between databases. In such cases, set this property to false. |
|  ClientCache | Gets the client page cache object |
|  DatabaseLocations | Gets the DatabaseLocations DatabaseLocations for the session. |
|  Databases | Gets a list of the currently opened databases |

| | |
|---------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DefaultCompressPages | By default false, set to true if you want to change the default for newly created DatabaseLocation . |
|  DefaultNumberOfObjectsPerPage | Default number of objects per page is 9999 |
|  DefaultStringComparer | Set the StringComparer to use when no application provided comparator is provided. By default we will then use StringComparer.Ordinal. |
|  DeletedOids | Collected Oid for unpersisted objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  DoWindowsAuthentication | By default do not use windows authentication when talking to VelocitydbServer's on other hosts. |
|  EmbedSerializableLists | Starting in version 7.1.6, List(T) bits are NOT by default stored with its parent object, instead each such List(T) is stored with its own object Id. This change was necessary in order to handle circular references via List(T) objects within an ISerializable serialization. Set this value to true if you have existing persisted ISerializable with List(T) or else we will try to read the lists as non embedded when they are not. |
|  EnableAutoPageFlush | Pages may be flushed when memory available is low unless you set tis property to false. You would want it to be enabled unless you are trying to share a non SessionNoServerShared session by working on different databases in multiple threads. As in Wikipedia sample application. |
|  EnableSyncByTrackingChanges | Enable database and page change tracking. Turn off within an update transaction so that prior change set can be deleted. |
|  FileBytesChunkSize | When transferring complete files from a server, we send the file bytes in chunks of this size. |
|  FileShareReader | FileShare for reader is ReadWrite with optimistic locking; otherwise Read |
|  FileShareUpdater | FileShare for updater is Read with optimistic locking; otherwise None |
|  InCommit | Indicates if we are within a transaction commit. |
|  InMemoryOnly | Gets information about this session, is it an in-memory only session or a session that persists data |
|  InTransaction | Indicates if we are within an active transaction. |

| | |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  InUpdateTransaction | Indicates if we are within an active update transaction. |
|  IsDisposed | Check if session has been disposed |
|  LocalHost | Get the host name of your local host (the machine you are using) |
|  MaxStringLengthToIntern | Get the maximum size a string must have before VelocityDB internals interns, Intern(String) , string when read from a Database . |
|  MaxUpdateTransactionLogHistory | Determines max number of UpdateTransaction objects we maintain in update transaction log in Database 0, initially set to 9990 |
|  MinStringLengthToIntern | Get the minimum size a string must have before VelocityDB internals interns, Intern(String) , string when read from a Database . |
|  NewDatabases | Get a list of all new databases created within this transaction |
|  NewOids | Collected Oid for newly persisted objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  NotifyBeforeCommit | Set to an Action of your choice if you want to get notified right before commit of an update transaction. |
|  ObjectCachingDefaultPolicy | Get/Set default value for Cache |
|  OptimisticLocking | Is the active Database and/or Page locking using optimistic locking model or not? See https://en.wikipedia.org/wiki/Optimistic_concurrency_control and https://en.wikipedia.org/wiki/Record_locking |
|  StrongReferenceDatabaseCaching | Is current session caching Databases using strong references or just by weak references. Set in session constructor or by calling CrossTransactionCacheAllDatabases(Boolean) . |
|  SystemDirectory | Get the path to the directory of the system Databases (0.oddb, 1.oddb, 2.oddb ...) of this session. |
|  SystemHostName | Get the host name of the system Databases (0.oddb, 1.oddb, 2.oddb ...) of this session. |
|  TextEncoding | All text encoding is done with utf8 |
|  TraceDBActivity | |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  TraceIndexUsage | If set to true, each time an index (BTreeSet) is used in a LINQ query expression, we output a line about that to the console output. |
|  TransactionNumber | Gets the transaction number of the current session |
|  UpdatedOids | Collected Oid for updated objects within an update transaction. Ids are only collected if NotifyBeforeCommit is non null. |
|  UseExternalStorageApi | This is specifically for Windows Phone API, set to true when you want to access databases from a memory card. |
|  WaitForLockMilliseconds | Get/Set the max time waiting for a lock to clear with pessimistic locking |
|  WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.AddToIndexInSeperateThread Property

Allow adding objects to indices to be done in a worker thread instead of in main thread.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool AddToIndexInSeperateThread { get; set; }
```

VB

```
Public Overridable Property AddToIndexInSeperateThread As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool AddToIndexInSeperateThread {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract AddToIndexInSeperateThread : bool with get, set  
override AddToIndexInSeperateThread : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.AssumeLocalHost Property

SessionNoServer ignores DatabaseLocation host value, it assumes localhost, ServerSessionClient does not assume local host, it follows the DatabaseLocation HostName value.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool AssumeLocalHost { get; }
```

VB

```
Public Overridable ReadOnly Property AssumeLocalHost As Boolean  
    Get
```

C++

```
public:  
virtual property bool AssumeLocalHost {  
    bool get ();  
}
```

F#

```
abstract AssumeLocalHost : bool with get  
override AssumeLocalHost : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.BaseDatabasePath Property

If specified boot path (system database path) in session constructor isn't an absolute path then this base path is prefixed to make an absolute path.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string BaseDatabasePath { get; set; }
```

VB

```
Public Shared Property BaseDatabasePath As String  
    Get  
    Set
```

C++

```
public:  
static property String^ BaseDatabasePath {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
static member BaseDatabasePath : string with get, set
```

Property Value

Type: [String](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.BTreeAddFastTransientBatchSize Property

Default batch size for BTree AddFast api, initially set to 100

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static uint BTreeAddFastTransientBatchSize { get; set; }
```

VB

```
Public Shared Property BTreeAddFastTransientBatchSize As UInteger  
    Get  
    Set
```

C++

```
public:  
static property unsigned int BTreeAddFastTransientBatchSize {  
    unsigned int get ();  
    void set (unsigned int value);  
}
```

F#

```
static member BTreeAddFastTransientBatchSize : uint32 with get, set
```

Property Value

Type: [UInt32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearAllCachedObjectsWhenDetectingUpdatedDatabase Property

By default we clear all cached object when at the beginning of a transaction we detect an updated [Database](#) within our cached databases. Same logic is applied when calling [ForceDatabaseCacheValidation\(\)](#). We do so because most of the time an object in one database have (strong) references to objects in other databases. If Database A is NOT updated but database B is and an object in database A is cached with a strong reference to an object in database B then our cached object in database A may contain a stale/incorrect reference to object in database B. This is why we, by default, invalidate all cached objects when we detect a modified database within our cached databases. However, if you design your database schema without strong references between databases then you do not need to invalidate cached objects for other databases, only the ones that have changed. By consistently using [WeakIOptimizedPersistableReference\(T\)](#), [BTreeSet\(Key\)](#) and [BTreeMap\(Key, Value\)](#), you can keep references weak between databases. In such cases, set this property to false.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool ClearAllCachedObjectsWhenDetectingUpdatedDatabase { get; set; }
```

VB

```
Public Shared Property ClearAllCachedObjectsWhenDetectingUpdatedDatabase As Boolean
    Get
    Set
```

C++

```
public:
    static property bool ClearAllCachedObjectsWhenDetectingUpdatedDatabase {
        bool get ();
        void set (bool value);
    }
```

F#

```
static member ClearAllCachedObjectsWhenDetectingUpdatedDatabase : bool with
    get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.ClientCache Property

Gets the client page cache object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DataCache ClientCache { get; }
```

VB

```
Public ReadOnly Property ClientCache As DataCache  
    Get
```

C++

```
public:  
property DataCache^ ClientCache {  
    DataCache^ get ();  
}
```

F#

```
member ClientCache : DataCache with get
```

Property Value

Type: [DataCache](#)

The [DataCache](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DatabaseLocations Property

Gets the DatabaseLocations **DatabaseLocations** for the session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocations DatabaseLocations { get; }
```

VB

```
Public ReadOnly Property DatabaseLocations As DatabaseLocations  
    Get
```

C++

```
public:  
property DatabaseLocations^ DatabaseLocations {  
    DatabaseLocations^ get ();  
}
```

F#

```
member DatabaseLocations : DatabaseLocations with get
```

Property Value

Type: [DatabaseLocations](#)

The **DatabaseLocations**

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Databases Property

Gets a list of the currently opened databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IList<Database> Databases { get; }
```

VB

```
Public Overridable ReadOnly Property Databases As IList(Of Database)  
    Get
```

C++

```
public:  
virtual property IList<Database^>^ Databases {  
    IList<Database^>^ get ();  
}
```

F#

```
abstract Databases : IList<Database> with get  
override Databases : IList<Database> with get
```

Property Value

Type: [IList\(Database\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DefaultCompressPages Property

By default false, set to true if you want to change the default for newly created [DatabaseLocation](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static PageInfo.compressionKind DefaultCompressPages { get; set; }
```

VB

```
Public Shared Property DefaultCompressPages As PageInfo.compressionKind  
    Get  
    Set
```

C++

```
public:  
static property PageInfo.compressionKind DefaultCompressPages {  
    PageInfo.compressionKind get ();  
    void set (PageInfo.compressionKind value);  
}
```

F#

```
static member DefaultCompressPages : PageInfo.compressionKind with get, set
```

Property Value

Type: [PageInfo.compressionKind](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DefaultNumberOfObjectsPerPage Property

Default number of objects per page is 9999

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort DefaultNumberOfObjectsPerPage { get; }
```

VB

```
Public Shared ReadOnly Property DefaultNumberOfObjectsPerPage As UShort  
    Get
```

C++

```
public:  
static property unsigned short DefaultNumberOfObjectsPerPage {  
    unsigned short get ();  
}
```

F#

```
static member DefaultNumberOfObjectsPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DefaultStringComparer Property

Set the StringComparer to use when no application provided comparator is provided. By default we will then use StringComparer.Ordinal.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static StringComparer DefaultStringComparer { get; set; }
```

VB

```
Public Shared Property DefaultStringComparer As StringComparer  
    Get  
    Set
```

C++

```
public:  
static property StringComparer^ DefaultStringComparer {  
    StringComparer^ get ();  
    void set (StringComparer^ value);  
}
```

F#

```
static member DefaultStringComparer : StringComparer with get, set
```

Property Value

Type: [StringComparer](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeletedOids Property

Collected [Oid](#) for unpersisted objects within an update transaction. Ids are only collected if [NotifyBeforeCommit](#) is non null.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public HashSet<Oid> DeletedOids { get; }
```

VB

```
Public ReadOnly Property DeletedOids As HashSet(Of Oid)  
    Get
```

C++

```
public:  
property HashSet<Oid>^ DeletedOids {  
    HashSet<Oid>^ get ();  
}
```

F#

```
member DeletedOids : HashSet<Oid> with get
```

Property Value

Type: [HashSet\(Oid\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DoWindowsAuthentication Property

By default do not use windows authentication when talking to VelocitydbServer's on other hosts.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool DoWindowsAuthentication { get; set; }
```

VB

```
Public Shared Property DoWindowsAuthentication As Boolean  
    Get  
    Set
```

C++

```
public:  
static property bool DoWindowsAuthentication {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
static member DoWindowsAuthentication : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.EmbedSerializableLists Property

Starting in version 7.1.6, [List\(T\)](#) bits are NOT by default stored with its parent object, instead each such [List\(T\)](#) is stored with its own object Id. This change was necessary in order to handle circular references via [List\(T\)](#) objects within an [ISerializable](#) serialization. Set this value to true if you have existing persisted [ISerializable](#) with [List\(T\)](#) or else we will try to read the lists as non embedded when they are not.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool EmbedSerializableLists { get; set; }
```

VB

```
Public Shared Property EmbedSerializableLists As Boolean  
    Get  
    Set
```

C++

```
public:  
static property bool EmbedSerializableLists {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
static member EmbedSerializableLists : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.EnableAutoPageFlush Property

Pages may be flushed when memory available is low unless you set tis property to false. You would want it to be enabled unless you are trying to share a non SessionNoServerShared session by working on different databases in multiple threads. As in Wikipedia sample application.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool EnableAutoPageFlush { get; set; }
```

VB

```
Public Property EnableAutoPageFlush As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool EnableAutoPageFlush {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member EnableAutoPageFlush : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.EnableSyncByTrackingChanges Property

Enable database and page change tracking. Turn off within an update transaction so that prior change set can be deleted.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool EnableSyncByTrackingChanges { get; set; }
```

VB

```
Public Property EnableSyncByTrackingChanges As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool EnableSyncByTrackingChanges {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member EnableSyncByTrackingChanges : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileBytesChunkSize Property

When transferring complete files from a server, we send the file bytes in chunks of this size.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static uint FileBytesChunkSize { get; }
```

VB

```
Public Shared ReadOnly Property FileBytesChunkSize As UInteger  
    Get
```

C++

```
public:  
static property unsigned int FileBytesChunkSize {  
    unsigned int get ();  
}
```

F#

```
static member FileBytesChunkSize : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileShareReader Property

[FileShare](#)

for reader is [ReadWrite](#) with optimistic locking; otherwise [Read](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FileShare FileShareReader { get; }
```

VB

```
Public ReadOnly Property FileShareReader As FileShare  
    Get
```

C++

```
public:  
property FileShare FileShareReader {  
    FileShare get ();  
}
```

F#

```
member FileShareReader : FileShare with get
```

Property Value

Type: [FileShare](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileShareUpdater Property

[FileShare](#)

for updater is [Read](#) with optimistic locking; otherwise [None](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FileShare FileShareUpdater { get; }
```

VB

```
Public ReadOnly Property FileShareUpdater As FileShare  
    Get
```

C++

```
public:  
property FileShare FileShareUpdater {  
    FileShare get ();  
}
```

F#

```
member FileShareUpdater : FileShare with get
```

Property Value

Type: [FileShare](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.InCommit Property

Indicates if we are within a transaction commit.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool InCommit { get; }
```

VB

```
Public Overridable ReadOnly Property InCommit As Boolean  
    Get
```

C++

```
public:  
virtual property bool InCommit {  
    bool get ();  
}
```

F#

```
abstract InCommit : bool with get  
override InCommit : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.InMemoryOnly Property

Gets information about this session, is it an in-memory only session or a session that persists data

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool InMemoryOnly { get; }
```

VB

```
Public ReadOnly Property InMemoryOnly As Boolean  
    Get
```

C++

```
public:  
property bool InMemoryOnly {  
    bool get ();  
}
```

F#

```
member InMemoryOnly : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.InTransaction Property

Indicates if we are within an active transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool InTransaction { get; }
```

VB

```
Public Overridable ReadOnly Property InTransaction As Boolean  
    Get
```

C++

```
public:  
virtual property bool InTransaction {  
    bool get ();  
}
```

F#

```
abstract InTransaction : bool with get  
override InTransaction : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.InUpdateTransaction Property

Indicates if we are within an active update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool InUpdateTransaction { get; }
```

VB

```
Public Overridable ReadOnly Property InUpdateTransaction As Boolean  
    Get
```

C++

```
public:  
virtual property bool InUpdateTransaction {  
    bool get ();  
}
```

F#

```
abstract InUpdateTransaction : bool with get  
override InUpdateTransaction : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.IsDisposed Property

Check if session has been disposed

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsDisposed { get; }
```

VB

```
Public ReadOnly Property IsDisposed As Boolean  
    Get
```

C++

```
public:  
property bool IsDisposed {  
    bool get ();  
}
```

F#

```
member IsDisposed : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.LocalHost Property

Get the host name of your local host (the machine you are using)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string LocalHost { get; }
```

VB

```
Public Shared ReadOnly Property LocalHost As String  
    Get
```

C++

```
public:  
static property String^ LocalHost {  
    String^ get ();  
}
```

F#

```
static member LocalHost : string with get
```

Property Value

Type: [String](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.MaxStringLengthToIntern Property

Get the maximum size a string must have before VelocityDB internals interns, [Intern\(String\)](#), string when read from a [Database](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort MaxStringLengthToIntern { get; }
```

VB

```
Public Shared ReadOnly Property MaxStringLengthToIntern As UShort  
    Get
```

C++

```
public:  
static property unsigned short MaxStringLengthToIntern {  
    unsigned short get ();  
}
```

F#

```
static member MaxStringLengthToIntern : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.MaxUpdateTransactionLogHistory Property

Determines max number of UpdateTransaction objects we maintain in update transaction log in Database 0, initially set to 9990

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int MaxUpdateTransactionLogHistory { get; set; }
```

VB

```
Public Shared Property MaxUpdateTransactionLogHistory As Integer  
    Get  
    Set
```

C++

```
public:  
static property int MaxUpdateTransactionLogHistory {  
    int get ();  
    void set (int value);  
}
```

F#

```
static member MaxUpdateTransactionLogHistory : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.MinStringLengthToIntern Property

Get the minimum size a string must have before VelocityDB internals interns, [Intern\(String\)](#), string when read from a [Database](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ushort MinStringLengthToIntern { get; }
```

VB

```
Public Shared ReadOnly Property MinStringLengthToIntern As UShort  
    Get
```

C++

```
public:  
static property unsigned short MinStringLengthToIntern {  
    unsigned short get ();  
}
```

F#

```
static member MinStringLengthToIntern : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.NewDatabases Property

Get a list of all new databases created within this transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<Database> NewDatabases { get; }
```

VB

```
Public ReadOnly Property NewDatabases As List(Of Database)  
    Get
```

C++

```
public:  
property List<Database^>^ NewDatabases {  
    List<Database^>^ get ();  
}
```

F#

```
member NewDatabases : List<Database> with get
```

Property Value

Type: [List\(Database\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.NewOids Property

Collected [Oid](#) for newly persisted objects within an update transaction. Ids are only collected if [NotifyBeforeCommit](#) is non null.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public HashSet<Oid> NewOids { get; }
```

VB

```
Public ReadOnly Property NewOids As HashSet(Of Oid)  
    Get
```

C++

```
public:  
property HashSet<Oid>^ NewOids {  
    HashSet<Oid>^ get ();  
}
```

F#

```
member NewOids : HashSet<Oid> with get
```

Property Value

Type: [HashSet\(Oid\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.NotifyBeforeCommit Property

Set to an Action of your choice if you want to get notified right before commit of an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Action<SessionBase> NotifyBeforeCommit { get; set; }
```

VB

```
Public Property NotifyBeforeCommit As Action(Of SessionBase)  
    Get  
    Set
```

C++

```
public:  
property Action<SessionBase^>^ NotifyBeforeCommit {  
    Action<SessionBase^>^ get ();  
    void set (Action<SessionBase^>^ value);  
}
```

F#

```
member NotifyBeforeCommit : Action<SessionBase> with get, set
```

Property Value

Type: [Action\(SessionBase\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ObjectCachingDefaultPolicy Property

Get/Set default value for [Cache](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public CacheEnum ObjectCachingDefaultPolicy { get; set; }
```

VB

```
Public Property ObjectCachingDefaultPolicy As CacheEnum  
    Get  
    Set
```

C++

```
public:  
property CacheEnum ObjectCachingDefaultPolicy {  
    CacheEnum get ();  
    void set (CacheEnum value);  
}
```

F#

```
member ObjectCachingDefaultPolicy : CacheEnum with get, set
```

Property Value

Type: [CacheEnum](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OptimisticLocking Property

Is the active [Database](#) and/or [Page](#) locking using optimistic locking model or not? See https://en.wikipedia.org/wiki/Optimistic_concurrency_control and https://en.wikipedia.org/wiki/Record_locking

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool OptimisticLocking { get; }
```

VB

```
Public ReadOnly Property OptimisticLocking As Boolean  
    Get
```

C++

```
public:  
property bool OptimisticLocking {  
    bool get ();  
}
```

F#

```
member OptimisticLocking : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.StrongReferenceDatabaseCaching Property

Is current session caching [Databases](#) using strong references or just by weak references. Set in session constructor or by calling [CrossTransactionCacheAllDatabases\(Boolean\)](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool StrongReferenceDatabaseCaching { get; }
```

VB

```
Public ReadOnly Property StrongReferenceDatabaseCaching As Boolean  
    Get
```

C++

```
public:  
property bool StrongReferenceDatabaseCaching {  
    bool get ();  
}
```

F#

```
member StrongReferenceDatabaseCaching : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SystemDirectory Property

Get the path to the directory of the system [Databases](#) (0.odt, 1.odt, 2.odt ...) of this session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SystemDirectory { get; }
```

VB

```
Public ReadOnly Property SystemDirectory As String  
    Get
```

C++

```
public:  
property String^ SystemDirectory {  
    String^ get ();  
}
```

F#

```
member SystemDirectory : string with get
```

Property Value

Type: [String](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SystemHostName Property

Get the host name of the system [Databases](#) (0.odb, 1.odb, 2.odb ...) of this session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SystemHostName { get; }
```

VB

```
Public ReadOnly Property SystemHostName As String  
    Get
```

C++

```
public:  
property String^ SystemHostName {  
    String^ get ();  
}
```

F#

```
member SystemHostName : string with get
```

Property Value

Type: [String](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.TextEncoding Property

All text encoding is done with utf8

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Encoding TextEncoding { get; }
```

VB

```
Public Shared ReadOnly Property TextEncoding As Encoding  
    Get
```

C++

```
public:  
static property Encoding^ TextEncoding {  
    Encoding^ get ();  
}
```

F#

```
static member TextEncoding : Encoding with get
```

Property Value

Type: [Encoding](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.TraceDBActivity Property

[Missing <summary> documentation for "P:VelocityDb.Session.SessionBase.TraceDBActivity"]

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SortedSet<uint> TraceDBActivity { get; }
```

VB

```
Public ReadOnly Property TraceDBActivity As SortedSet(Of UInteger)  
    Get
```

C++

```
public:  
property SortedSet<unsigned int>^ TraceDBActivity {  
    SortedSet<unsigned int>^ get ();  
}
```

F#

```
member TraceDBActivity : SortedSet<uint32> with get
```

Property Value

Type: [SortedSet\(UInt32\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.TraceIndexUsage Property

If set to true, each time an index (BTreeSet) is used in a LINQ query expression, we output a line about that to the console output.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TraceIndexUsage { get; set; }
```

VB

```
Public Property TraceIndexUsage As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool TraceIndexUsage {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member TraceIndexUsage : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.TransactionNumber Property

Gets the transaction number of the current session

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong TransactionNumber { get; }
```

VB

```
Public ReadOnly Property TransactionNumber As ULong  
    Get
```

C++

```
public:  
property unsigned long long TransactionNumber {  
    unsigned long long get ();  
}
```

F#

```
member TransactionNumber : uint64 with get
```

Property Value

Type: [UInt64](#)

The transaction number

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdatedOids Property

Collected [Oid](#) for updated objects within an update transaction. Ids are only collected if [NotifyBeforeCommit](#) is non null.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public HashSet<Oid> UpdatedOids { get; }
```

VB

```
Public ReadOnly Property UpdatedOids As HashSet(Of Oid)  
    Get
```

C++

```
public:  
property HashSet<Oid>^ UpdatedOids {  
    HashSet<Oid>^ get ();  
}
```

F#

```
member UpdatedOids : HashSet<Oid> with get
```

Property Value

Type: [HashSet\(Oid\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UseExternalStorageApi Property

This is specifically for Windows Phone API, set to true when you want to access databases from a memory card.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool UseExternalStorageApi { get; set; }
```

VB

```
Public Property UseExternalStorageApi As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool UseExternalStorageApi {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member UseExternalStorageApi : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.WaitForLockMilliseconds Property

Get/Set the max time waiting for a lock to clear with pessimistic locking

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int WaitForLockMilliseconds { get; set; }
```

VB

```
Public Property WaitForLockMilliseconds As Integer  
    Get  
    Set
```

C++

```
public:  
property int WaitForLockMilliseconds {  
    int get ();  
    void set (int value);  
}
```

F#

```
member WaitForLockMilliseconds : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.WriteToDiskInSeperateDatabaseThreads Property

Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. [AddToIndexInSeperateThread](#) must also be enabled for this to work.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool WriteToDiskInSeperateDatabaseThreads { get; set; }
```

VB

```
Public Overridable Property WriteToDiskInSeperateDatabaseThreads As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool WriteToDiskInSeperateDatabaseThreads {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract WriteToDiskInSeperateDatabaseThreads : bool with get, set  
override WriteToDiskInSeperateDatabaseThreads : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionBase Class](#)













[VelocityDb.Session Namespace](#)




















SessionBase.SessionBase Methods
























The [SessionBase](#) type exposes the following members.











Methods



















| Name | Description |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Abort | Transaction control, abort the current transaction |
| AllObjects(T) | Gets an object used for enumerating all objects in all Databases |
| AnyPageWritten | Find out if any pages have been written for a given database in this update transaction |
| BeginRead | Transaction control, begin a read only transaction |
| BeginUpdate() | Transaction control, begin an update transaction. |
| BeginUpdate(Boolean) | Transaction control, begin an update transaction. |
| Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() |
| ClearCache | Clears page cache and closes databases. Avoid using this one for now. |
| ClearCachedObjects() | Clears the cache of cached objects. |
| ClearCachedObjects(Database) | Clears the cache of cached objects within a Database . |
| ClearCachedPages | Clear cache of cached pages |
| ClearPageCache | Clears cached pages from cache including page weak references. |
| CloseDatabase | Closes a Database |
| Commit | Transaction control, commit the current transaction |
| Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. |
| Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins |
| CompareTo | The default compare of two SessionBase is by system (bootup) host name and directory |
| ContainsDatabase | Check if database file exist in a given DatabaseLocation |
| CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. |
| CreateDirectory | Creates a directory |
| CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This |





| | | |
|-------------------------------------------------------------------------------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened. |
|  | CrossTransactionCacheAllDatabases | By default databases are referenced by strong references instead of WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to all Databases so the cached Database may be used if version wasn't changed by a different thread since prior transaction. A strong reference is removed for any database that get invalidated due to a transaction abort or if another transaction commits a change to it. Check current session state with StrongReferenceDatabaseCaching |
|  | DatabaseNumberOf | Get the database number associated with a specific type. Your application can store any type of object in any database but if the application exclusively use Persist(IOptimizedPersistable, Nullable(UInt16)) or Persist(Object) then this is the database Id that will be used given a type. |
|  | DatabaseStillExist | Check if Database still exist |
|  | DefaultDatabaseLocation | Gets the default (startup) DatabaseLocation |
|  | DeleteDatabase | Delete a database, actual database file delete happens after a successful transaction commit. |
|  | DeleteFile | Delete a file. For internal use. |
|  | DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it |
|  | DeleteObject | A way to delete an object without requiring that the object be opened first |
|  | DeletePage | Deletes a page from a database. (part of database Sync beta version - not yet fully designed and tested) Used by SyncWith when syncing databases with another replica of the current databases |
|  | DeployGenerateReaderWriter | Part of code generator, not yet ready for public release |
|  | DeployInternalTypes | For internal use only |
|  | Dispose | Closes this session |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | Opens a Database file for read/update. Internal use only. |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | For internal use. |
|  FirstFreePage | Gets the page number of the first unallocated page in the specified Database |
|  FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written. |
|  FlushUpdates() | Write all updated data. This may free up some memory. |
|  FlushUpdates(Database) | Write any updated/new pages for a specific Database |
|  FlushUpdatesServers | Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s) |
|  ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. |
|  GetEnumerator | Enumerates all open databases for this session |
|  GetVersion | Gets the version of Database 0 |
|  GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object |
|  HighestInUseDatabaseNumber | Figure out what the highest in use database number is and return it. |
|  HighestInUseLocalDatabaseNumber | Figure out what the highest in use database number is on the local host and return it. |
|  IdOf | Finds the object id of a persistent object |
|  Index(T)() | Gets an index of all objects of a certain type |
|  Index(T)(String) | Gets an index of all objects of a certain type |
|  Index(T)(Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used). If you have updated objects persisted in a prior transaction that you want part of an index. Call FlushUpdates() or call Write()/> to add such an object to index before querying using an index./> |
|  Index(T)(String, Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used) |
|  InUseNumberOfPages | Gets the number of pages currently in use by a Database |










| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IsSameHost  | Determine if two host names represent the same tcp/ip host |
|  IsWorkerThread | Tell if code is executed by index or page write worker thread |
|  LoadFields | Loads all field values of an object if they are not already loaded. |
|  LocateDb | Lookup the DatabaseLocation of a Database with a specified database number. |
|  NewDatabase | Create a new Database with a given database number |
|  NewLocation | Creates a new DatabaseLocation or updates existing ones |
|  Objects(T) | Enumerates all the objects of a given type |
|  OfType | Gets an object used for enumerating all objects in all Databases |
|  OidOf | Finds the object id of a persistent object |
|  OidShortOf | Finds the object id of a persistent object |
|  OpenAllDatabases | Open all databases |
|  OpenDatabase | Opens a Database |
|  OpenDatabaseLocations | Open the DatabaseLocations object and store it in the session. |
|  OpenLocationDatabases | Opens all the databases in a given location |
|  OpenPage | Opens a page for read, used by DatabaseManager for page browsing |
|  OpenSchema | Get the session active schema |
|  ParseCsvRow  | Parses a row in a csv file and returns an iterator of column string values |
|  Persist(Object) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database. |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . |
|  Persist(Placement, IOptimizedPersistable) | Persist an object that is an OptimizedPersistable or a subclass |
|  Persist(Object, IOptimizedPersistable, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Persist(Object, Placement, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. |
|  RegisterClass | Register a class in the persistent schema. It is a good idea to do this explicitly instead of having it happen the first time an object of an unregistered class is made persistent |
|  RelocateDatabaseLocationFor | After moving/copying directory containing your Database with id dbId call this before starting a transaction to update host and directory of the associated DatabaseLocation . This function does not move the database files, it only updates the DatabaseLocation . |
|  RelocateDefaultDatabaseLocation | After moving/copying directory containing your Databases 0.odb, 1.odb ... call this before starting a transaction to update host and directory of the default DatabaseLocation . This function does not move the database files, it only updates the DatabaseLocation . |
|  ReplacePage | Only to be used in VelocityDbExtensions while syncing databases |
|  ReplacePersistedType(String, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old Type loaded. Be sure to backup all your data before doing this. |
|  ReplacePersistedType(Type, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old Type loaded. Be sure to backup all your data before doing this. |
|  RestoreFrom(DatabaseLocation, DateTime) | Restores Databases and pages from a backup DatabaseLocation . Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  RestoreFrom(String, String, UInt32, DateTime) | Restores a DatabaseLocation |
|   SetMinMaxStringIntern | A persistent string read into memory from a Database is interned using Intern(String) to speed up comparisons and conserve memory (when the same string occurs in many places) |
|  SetTraceAllDbActivity | Enable some Trace output about persistent events related to all databases and pages. Add a Trace listener to get to Console window: <code>Trace.Listeners.Add(new ConsoleTraceListener());</code> |
|  SetTraceDbActivity | Enable some Trace output about persistent events related to a Database. Add a Trace listener to get to Console window: <code>Trace.Listeners.Add(new ConsoleTraceListener());</code> |
|   StringToType | Gets a Type given a string representation of a type |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. |
|  TryDatabase | Gets the current database to try |
|  Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable |
|  UnsubscribeToChanges | Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true. |
|  UpdateClass | Let VelocityDB know that changes have been made to a class so that the schema manager will create a new TypeVersion and use the new version for all new objects of the specified Type. |
|  UpdateDatabase | Request an update lock on a database |
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently |
|  UpdateObject(IOptimizedPersistable) | Tag an object as updated so that it will be updated persistently |
|  UpdateObject(Object, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
|  UpdateObject(IOptimizedPersistable, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object |

| | |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
|  UpdatePage | Update a Page . Used by VelocityDBExtensions project |
|  Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. |
|  WaitForIndexUpdates | Internal Use and in VelocityDBExtensions |
|  WritePageBytes | |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[SessionBase Class](#)[VelocityDb.Session Namespace](#)

SessionBase.Abort Method

Transaction control, abort the current transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Abort ()
```

VB

```
Public Overridable Sub Abort
```

C++

```
public:  
virtual void Abort ()
```

F#

```
abstract Abort : unit -> unit  
override Abort : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.AllObjects(T) Method

Gets an object used for enumerating all objects in all Databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AllObjects<T> AllObjects<T>(
    bool includeSubclasses = true,
    bool databasePerType = true
)
```

VB

```
Public Function AllObjects(Of T) (
    Optional includeSubclasses As Boolean = true,
    Optional databasePerType As Boolean = true
) As AllObjects(Of T)
```

C++

```
public:
generic<typename T>
AllObjects<T>^ AllObjects(
    bool includeSubclasses = true,
    bool databasePerType = true
)
```

F#

```
member AllObjects :
    ?includeSubclasses : bool *
    ?databasePerType : bool
(* Defaults:
    let _includeSubclasses = defaultArg includeSubclasses true
    let _databasePerType = defaultArg databasePerType true
*)
-> AllObjects<'T>
```

Parameters

includeSubclasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes or classes that implements the specified interface class

databasePerType (Optional)

Type: [System.Boolean](#)

Assume that persisted objects were made persistent the simple way using [Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#). When persisting this way, each object type gets its own [Database](#) which makes finding these objects easier and faster

Type Parameters

T

The type of object we are looking for

Return Value

Type: [AllObjects\(T\)](#)

The enumeration wrapper object

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.AnyPageWritten Method

Find out if any pages have been written for a given database in this update transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AnyPageWritten(  
    Database db  
)
```

VB

```
Public Function AnyPageWritten (  
    db As Database  
) As Boolean
```

C++

```
public:  
bool AnyPageWritten(  
    Database^ db  
)
```

F#

```
member AnyPageWritten :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database of interest

Return Value

Type: [Boolean](#)

true if any pages have been written, otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.BeginRead Method

Transaction control, begin a read only transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual SessionBase.Transaction BeginRead(  
    bool doRecoveryCheck = false  
)
```

VB

```
Public Overridable Function BeginRead (  
    Optional doRecoveryCheck As Boolean = false  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginRead(  
    bool doRecoveryCheck = false  
)
```

F#

```
abstract BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction  
override BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

By default we do not check for a failed update transaction that requires reverting one or more database changes. We don't do it by default because it can be a little costly. Change this parameter to `true` if you prefer ensured consistency over better performance or make change only if you encounter an exception in the transaction.

VelocityDB Class Library

Return Value

Type: [SessionBase.Transaction](#)

A transaction helper object.



See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.BeginUpdate Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------|
|  | BeginUpdate() | Transaction control, begin an update transaction. |
|  | BeginUpdate(Boolean) | Transaction control, begin an update transaction. |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.BeginUpdate Method

Transaction control, begin an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionBase.Transaction BeginUpdate ()
```

VB

```
Public Function BeginUpdate As SessionBase.Transaction
```

C++

```
public:  
SessionBase.Transaction^ BeginUpdate ()
```

F#

```
member BeginUpdate : unit -> SessionBase.Transaction
```

Return Value

Type: [SessionBase.Transaction](#)

A [SessionBase.Transaction](#) helper object.

See Also

[SessionBase Class](#)

[BeginUpdate Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.BeginUpdate Method (Boolean)

Transaction control, begin an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual SessionBase.Transaction BeginUpdate(  
    bool doRecoveryCheck  
)
```

VB

```
Public Overridable Function BeginUpdate (  
    doRecoveryCheck As Boolean  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginUpdate(  
    bool doRecoveryCheck  
)
```

F#

```
abstract BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction  
override BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction
```

Parameters

doRecoveryCheck

Type: [System.Boolean](#)

Set to false only when moving system databases to a new directory

Return Value

Type: [SessionBase.Transaction](#)

A [SessionBase.Transaction](#) helper object.

See Also

[SessionBase Class](#)

[BeginUpdate Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Checkpoint Method

Same as [Commit\(Boolean, Boolean\)](#) followed by [BeginUpdate\(\)](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual SessionBase.Transaction Checkpoint ()
```

VB

```
Public Overridable Function Checkpoint As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ Checkpoint ()
```

F#

```
abstract Checkpoint : unit -> SessionBase.Transaction  
override Checkpoint : unit -> SessionBase.Transaction
```

Return Value

Type: [SessionBase.Transaction](#)

[SessionBase.Transaction](#) started

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearCache Method

Clears page cache and closes databases. Avoid using this one for now.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ClearCache ()
```

VB

```
Public Overridable Sub ClearCache
```

C++

```
public:  
virtual void ClearCache ()
```

F#

```
abstract ClearCache : unit -> unit  
override ClearCache : unit -> unit
```



See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearCachedObjects Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|------------------------------------------------------------------------|
|  | ClearCachedObjects() | Clears the cache of cached objects. |
|  | ClearCachedObjects(Database) | Clears the cache of cached objects within a Database . |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearCachedObjects Method

Clears the cache of cached objects.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ClearCachedObjects ()
```

VB

```
Public Sub ClearCachedObjects
```

C++

```
public:  
void ClearCachedObjects ()
```

F#

```
member ClearCachedObjects : unit -> unit
```

See Also

[SessionBase Class](#)

[ClearCachedObjects Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearCachedObjects Method (Database)

Clears the cache of cached objects within a [Database](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public void ClearCachedObjects (  
    Database db  
)
```

```
VB  
Public Sub ClearCachedObjects (  
    db As Database  
)
```

```
C++  
public:  
void ClearCachedObjects (  
    Database^ db  
)
```

```
F#  
member ClearCachedObjects :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

Clear cached objects within this database.

See Also

[SessionBase Class](#)

[ClearCachedObjects Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearCachedPages Method

Clear cache of cached pages

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ClearCachedPages ()
```

VB

```
Public Overridable Sub ClearCachedPages
```

C++

```
public:  
virtual void ClearCachedPages ()
```

F#

```
abstract ClearCachedPages : unit -> unit  
override ClearCachedPages : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ClearPageCache Method

Clears cached pages from cache including page weak references.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ClearPageCache ()
```

VB

```
Public Overridable Sub ClearPageCache
```

C++

```
public:  
virtual void ClearPageCache ()
```

F#

```
abstract ClearPageCache : unit -> unit  
override ClearPageCache : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.CloseDatabase Method

Closes a Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CloseDatabase (  
    Database db  
)
```

VB

```
Public Sub CloseDatabase (  
    db As Database  
)
```

C++

```
public:  
void CloseDatabase (  
    Database^ db  
)
```

F#

```
member CloseDatabase :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

The Database to close

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Commit Method

Transaction control, commit the current transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit(  
    bool doRecoveryCheck = true,  
    bool isRestore = false  
)
```

VB

```
Public Sub Commit (  
    Optional doRecoveryCheck As Boolean = true,  
    Optional isRestore As Boolean = false  
)
```

C++

```
public:  
void Commit(  
    bool doRecoveryCheck = true,  
    bool isRestore = false  
)
```

F#

```
member Commit :  
    ?doRecoveryCheck : bool *  
    ?isRestore : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck true  
    let _isRestore = defaultArg isRestore false  
)  
-> unit
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

Only case for not doing it is when committing a [RestoreFrom\(String, String, UInt32, DateTime\)](#)

isRestore (Optional)

Type: [System.Boolean](#)

Inform session that this is part of a restore

See Also



[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.Compact Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. |
|  | Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Compact Method

Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each [Database](#) file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Compact ()
```

VB

```
Public Overridable Sub Compact
```

C++

```
public:  
virtual void Compact ()
```

F#

```
abstract Compact : unit -> unit  
override Compact : unit -> unit
```

See Also

[SessionBase Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Compact Method (Database)

Reduce size of database, if possible, by truncating file where unused space begins

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Compact (  
    Database db  
)
```

VB

```
Public Overridable Sub Compact (  
    db As Database  
)
```

C++

```
public:  
virtual void Compact (  
    Database^ db  
)
```

F#

```
abstract Compact :  
    db : Database -> unit  
override Compact :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

to compact

See Also

[SessionBase Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.CompareTo Method

The default compare of two SessionBase is by system (bootup) host name and directory

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    SessionBase otherSession  
)
```

VB

```
Public Function CompareTo (  
    otherSession As SessionBase  
) As Integer
```

C++

```
public:  
int CompareTo(  
    SessionBase^ otherSession  
)
```

F#

```
member CompareTo :  
    otherSession : SessionBase -> int
```

Parameters

otherSession

Type: [VelocityDb.Session.SessionBase](#)

The session to compare with

Return Value

Type: [Int32](#)

-1 if this session is less than the other, 0 if equal, otherwise 1

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ContainsDatabase Method

Check if database file exist in a given [DatabaseLocation](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool ContainsDatabase(  
    DatabaseLocation location,  
    uint dbNum,  
    string extension = ".odb"  
)
```

VB

```
Public Overridable Function ContainsDatabase (  
    location As DatabaseLocation,  
    dbNum As UInteger,  
    Optional extension As String = ".odb"  
) As Boolean
```

C++

```
public:  
virtual bool ContainsDatabase(  
    DatabaseLocation^ location,  
    unsigned int dbNum,  
    String^ extension = L".odb"  
)
```

F#

```
abstract ContainsDatabase :  
    location : DatabaseLocation *  
    dbNum : uint32 *  
    ?extension : string  
(* Defaults:  
    let _extension = defaultArg extension ".odb"  
*)  
-> bool  
override ContainsDatabase :  
    location : DatabaseLocation *  
    dbNum : uint32 *  
    ?extension : string  
(* Defaults:  
    let _extension = defaultArg extension ".odb"  
*)  
-> bool
```

Parameters

location

VelocityDB Class Library

Type: [VelocityDb.DatabaseLocation](#)

Location to check

dbNum

Type: [System.UInt32](#)

Database number/id

extension (Optional)

Type: [System.String](#)

File name extension/id

Return Value

Type: [Boolean](#)

true if exist; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.CopyAllDatabasesTo Method

Copies all databases to a selected directory on the local host.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void CopyAllDatabasesTo(  
    string directory,  
    bool systemDatabaseLocationOnly = false  
)
```

VB

```
Public Overridable Sub CopyAllDatabasesTo (  
    directory As String,  
    Optional systemDatabaseLocationOnly As Boolean = false  
)
```

C++

```
public:  
virtual void CopyAllDatabasesTo(  
    String^ directory,  
    bool systemDatabaseLocationOnly = false  
)
```

F#

```
abstract CopyAllDatabasesTo :  
    directory : string *  
    ?systemDatabaseLocationOnly : bool  
(* Defaults:  
    let_systemDatabaseLocationOnly = defaultArg  
systemDatabaseLocationOnly false  
)  
-> unit  
override CopyAllDatabasesTo :  
    directory : string *  
    ?systemDatabaseLocationOnly : bool  
(* Defaults:  
    let_systemDatabaseLocationOnly = defaultArg  
systemDatabaseLocationOnly false  
)  
-> unit
```

Parameters

directory

Type: [System.String](#)

Path to a directory

VelocityDB Class Library

systemDatabaseLocationOnly (Optional)

Type: [System.Boolean](#)

[Missing <param name="systemDatabaseLocationOnly"/> documentation for "M:VelocityDb.Session.SessionBase.CopyAllDatabasesTo(System.String,System.Boolean)"]

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.CreateDirectory Method

Creates a directory

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool CreateDirectory(  
    string path  
)
```

VB

```
Public Overridable Function CreateDirectory (  
    path As String  
) As Boolean
```

C++

```
public:  
virtual bool CreateDirectory(  
    String^ path  
)
```

F#

```
abstract CreateDirectory :  
    path : string -> bool  
override CreateDirectory :  
    path : string -> bool
```

Parameters

path

Type: [System.String](#)

Return Value

Type: [Boolean](#)

true if path exist; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.CrossTransactionCache Method

By default databases are only referenced by a [WeakReference](#) across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void CrossTransactionCache (
    Database db,
    bool enable = true
)
```

VB

```
Public Overridable Sub CrossTransactionCache (
    db As Database,
    Optional enable As Boolean = true
)
```

C++

```
public:
virtual void CrossTransactionCache (
    Database^ db,
    bool enable = true
)
```

F#

```
abstract CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
override CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
```

Parameters

db

VelocityDB Class Library

Type: [VelocityDb.Database](#)

The Database to cache

enable (Optional)

Type: [System.Boolean](#)

Add or remove strong reference. If true, add a strong reference

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.CrossTransactionCacheAllDatabases Method

By default databases are referenced by strong references instead of [WeakReference](#) across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to all Databases so the cached Database may be used if version wasn't changed by a different thread since prior transaction. A strong reference is removed for any database that get invalidated due to a transaction abort or if another transaction commits a change to it. Check current session state with [StrongReferenceDatabaseCaching](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CrossTransactionCacheAllDatabases (  
    bool enable = true  
)
```

VB

```
Public Sub CrossTransactionCacheAllDatabases (  
    Optional enable As Boolean = true  
)
```

C++

```
public:  
void CrossTransactionCacheAllDatabases (  
    bool enable = true  
)
```

F#

```
member CrossTransactionCacheAllDatabases :  
    ?enable : bool  
(* Defaults:  
    let _enable = defaultArg enable true  
*)  
-> unit
```

Parameters

enable (Optional)

Type: [System.Boolean](#)

Add or remove strong reference. If true, add a strong reference

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DatabaseNumberOf Method

Get the database number associated with a specific type. Your application can store any type of object in any database but if the application exclusively use [Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#) or [Persist\(Object\)](#) then this is the database Id that will be used given a type.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint DatabaseNumberOf (
    Type type
)
```

VB

```
Public Function DatabaseNumberOf (
    type As Type
) As UInteger
```

C++

```
public:
    unsigned int DatabaseNumberOf (
        Type^ type
    )
```

F#

```
member DatabaseNumberOf :
    type : Type -> uint32
```

Parameters

type

Type: [System.Type](#)

The type for which you want to get the associated database number (Id)

Return Value

Type: [UInt32](#)

The database number associated with the provided type

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DatabaseStillExist Method

Check if [Database](#) still exist

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool DatabaseStillExist(  
    Database db  
)
```

VB

```
Public Overridable Function DatabaseStillExist (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool DatabaseStillExist(  
    Database^ db  
)
```

F#

```
abstract DatabaseStillExist :  
    db : Database -> bool  
override DatabaseStillExist :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

to check

Return Value

Type: [Boolean](#)

true if exists; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DefaultDatabaseLocation Method

Gets the default (startup) DatabaseLocation

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual DatabaseLocation DefaultDatabaseLocation()
```

VB

```
Public Overridable Function DefaultDatabaseLocation As DatabaseLocation
```

C++

```
public:  
virtual DatabaseLocation^ DefaultDatabaseLocation()
```

F#

```
abstract DatabaseLocation : unit -> DatabaseLocation  
override DatabaseLocation : unit -> DatabaseLocation
```

Return Value

Type: [DatabaseLocation](#)

The [DatabaseLocation](#) containing at least the system [Databases](#) (0.odb, 1.odb, 2.odb, 3.odb and 4.odb)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeleteDatabase Method

Delete a database, actual database file delete happens after a successful transaction commit.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeleteDatabase (  
    Database db  
)
```

VB

```
Public Sub DeleteDatabase (  
    db As Database  
)
```

C++

```
public:  
void DeleteDatabase (  
    Database^ db  
)
```

F#

```
member DeleteDatabase :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

The [Database](#) to delete

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeleteFile Method

Delete a file. For internal use.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public virtual void DeleteFile(  
    FileInfo fileInfo  
)
```

```
VB  
Public Overridable Sub DeleteFile (  
    fileInfo As FileInfo  
)
```

```
C++  
public:  
virtual void DeleteFile(  
    FileInfo^ fileInfo  
)
```

```
F#  
abstract DeleteFile :  
    fileInfo : FileInfo -> unit  
override DeleteFile :  
    fileInfo : FileInfo -> unit
```

Parameters

fileInfo

Type: [System.IO.FileInfo](#)

info about file to delete

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeleteLocation Method

Deletes a [DatabaseLocation](#), location must first not have any Databases in it

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void DeleteLocation(  
    DatabaseLocation location,  
    bool catalogOnly = false  
)
```

VB

```
Public Overridable Sub DeleteLocation (  
    location As DatabaseLocation,  
    Optional catalogOnly As Boolean = false  
)
```

C++

```
public:  
virtual void DeleteLocation(  
    DatabaseLocation^ location,  
    bool catalogOnly = false  
)
```

F#

```
abstract DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
*)  
-> unit  
override DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
*)  
-> unit
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The DatabaseLocation to delete

catalogOnly (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Of true, only delete catalog entry. Leave the directory as is.

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeleteObject Method

A way to delete an object without requiring that the object be opened first

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeleteObject (  
    ulong id  
)
```

VB

```
Public Sub DeleteObject (  
    id As ULong  
)
```

C++

```
public:  
void DeleteObject (  
    unsigned long long id  
)
```

F#

```
member DeleteObject :  
    id : uint64 -> unit
```

Parameters

id

Type: [System.UInt64](#)

The Id of the object to delete

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeletePage Method

Deletes a page from a database. (part of database Sync beta version - not yet fully designed and tested)
Used by SyncWith when syncing databases with another replica of the current databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeletePage(  
    Database db,  
    Page page  
)
```

VB

```
Public Sub DeletePage (  
    db As Database,  
    page As Page  
)
```

C++

```
public:  
void DeletePage(  
    Database^ db,  
    Page^ page  
)
```

F#

```
member DeletePage :  
    db : Database *  
    page : Page -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

Database containing page to delete

page

Type: [VelocityDb.Page](#)

Page to delete

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeployGenerateReaderWriter Method

Part of code generator, not yet ready for public release

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeployGenerateReaderWriter (
    Type type,
    string directory
)
```

VB

```
Public Sub DeployGenerateReaderWriter (
    type As Type,
    directory As String
)
```

C++

```
public:
void DeployGenerateReaderWriter (
    Type^ type,
    String^ directory
)
```

F#

```
member DeployGenerateReaderWriter :
    type : Type *
    directory : string -> unit
```

Parameters

type

Type: [System.Type](#)

Type to generate reader and writer code for

directory

Type: [System.String](#)

A directory for generated code

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.DeployInternalTypes Method

For internal use only

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DeployInternalTypes (  
    string outputDirectory  
)
```

VB

```
Public Sub DeployInternalTypes (  
    outputDirectory As String  
)
```

C++

```
public:  
void DeployInternalTypes (  
    String^ outputDirectory  
)
```

F#

```
member DeployInternalTypes :  
    outputDirectory : string -> unit
```

Parameters

outputDirectory

Type: [System.String](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Dispose Method

Closes this session

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)



See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileOpen Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|  | FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | Opens a Database file for read/update. Internal use only. |
|  | FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | For internal use. |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileOpen Method (Database, FileAccess, String, FileMode, Boolean, Int32, Boolean)

Opens a [Database](#) file for read/update. Internal use only.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Stream FileOpen(  
    Database db,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileMode fileMode = FileMode.Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
)
```

VB

```
Public Overridable Function FileOpen (  
    db As Database,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    Optional fileMode As FileMode = FileMode.Open,  
    Optional exclusiveAccess As Boolean = false,  
    Optional waitOverride As Integer = -1,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    Database^ db,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileMode fileMode = FileMode::Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
)
```

F#

```
abstract FileOpen :  
    db : Database *  
    fileAccess : FileAccess *  
    errorMessage : string byref *  
    ?fileMode : FileMode *  
    ?exclusiveAccess : bool *
```

```
        ?waitOverride : int *
        ?signalError : bool
(* Defaults:
    let_fileMode = defaultArg fileMode FileMode.Open
    let_exclusiveAccess = defaultArg exclusiveAccess false
    let_waitOverride = defaultArg waitOverride -1
    let_signalError = defaultArg signalError true
*)
-> Stream
override FileOpen :
    db : Database *
    fileAccess : FileAccess *
    errorMessage : string byref *
    ?fileMode : FileMode *
    ?exclusiveAccess : bool *
    ?waitOverride : int *
    ?signalError : bool
(* Defaults:
    let_fileMode = defaultArg fileMode FileMode.Open
    let_exclusiveAccess = defaultArg exclusiveAccess false
    let_waitOverride = defaultArg waitOverride -1
    let_signalError = defaultArg signalError true
*)
-> Stream
```

Parameters

db

Type: [VelocityDb.Database](#)
[Database](#)

for which we want to open [Stream](#)

fileAccess

Type: [System.IO.FileAccess](#)
Opening for read or update?

errorMessage

Type: [System.String](#)
Used for passing error messages back to caller

fileMode (Optional)

Type: [System.IO.FileMode](#)
[FileMode](#)

used for opening file

exclusiveAccess (Optional)

Type: [System.Boolean](#)
Do we want exclusive access to file?

waitOverride (Optional)

Type: [System.Int32](#)

VelocityDB Class Library

How long are we willing to wait for a file lock to clear?

signalError (Optional)

Type: [System.Boolean](#)

Throw exception if we can't open file within wait time?

Return Value

Type: [Stream](#)

A [Stream](#) used for accessing [Database](#) data

See Also

[SessionBase Class](#)

[FileOpen Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.FileOpen Method (FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean)

For internal use.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Stream FileOpen(  
    FileInfo fileInfo,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
)
```

VB

```
Public Overridable Function FileOpen (  
    fileInfo As FileInfo,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    fileShare As FileShare,  
    fileMode As FileMode,  
    waitForLockMilliseconds As Integer,  
    useExternalStorage As Boolean,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    FileInfo^ fileInfo,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
)
```

F#

```
abstract FileOpen :  
    fileInfo : FileInfo *  
    fileAccess : FileAccess *
```

```
        errorMessage : string byref *
        fileShare : FileShare *
        fileMode : FileMode *
        waitForLockMilliseconds : int *
        useExternalStorage : bool *
        ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream
override FileOpen :
    fileInfo : FileInfo *
    fileAccess : FileAccess *
    errorMessage : string byref *
    fileShare : FileShare *
    fileMode : FileMode *
    waitForLockMilliseconds : int *
    useExternalStorage : bool *
    ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream
```

Parameters

fileInfo

Type: [System.IO.FileInfo](#)

Info about file to open

fileAccess

Type: [System.IO.FileAccess](#)

Option for file access

errorMessage

Type: [System.String](#)

Error messages stored in this string

fileShare

Type: [System.IO.FileShare](#)

Option for file share

fileMode

Type: [System.IO.FileMode](#)

Option for file mode

waitForLockMilliseconds

Type: [System.Int32](#)

How long to wait for file to become available for open

useExternalStorage

Type: [System.Boolean](#)

VelocityDB Class Library

Parameter used for WindowsPhone API for accessing memory cards

signalError (Optional)

Type: [System.Boolean](#)

If true, signal error if we fail to open file; otherwise ignore errors and return null

Return Value

Type: [Stream](#)

File stream of opened file

See Also

[SessionBase Class](#)

[FileOpen Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.FirstFreePage Method

Gets the page number of the first unallocated page in the specified Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ushort FirstFreePage(  
    Database db,  
    ushort pageNumber  
)
```

VB

```
Public Overridable Function FirstFreePage (  
    db As Database,  
    pageNumber As UShort  
) As UShort
```

C++

```
public:  
virtual unsigned short FirstFreePage(  
    Database^ db,  
    unsigned short pageNumber  
)
```

F#

```
abstract FirstFreePage :  
    db : Database *  
    pageNumber : uint16 -> uint16  
override FirstFreePage :  
    db : Database *  
    pageNumber : uint16 -> uint16
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to query

pageNumber

Type: [System.UInt16](#)

The page number to start search from

Return Value

Type: [UInt16](#)

The first available page number.

VelocityDB Class Library

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FlushPageOf Method

By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void FlushPageOf(  
    OptimizedPersistable pObj  
)
```

VB

```
Public Overridable Sub FlushPageOf (  
    pObj As OptimizedPersistable  
)
```

C++

```
public:  
virtual void FlushPageOf(  
    OptimizedPersistable^ pObj  
)
```

F#

```
abstract FlushPageOf :  
    pObj : OptimizedPersistable -> unit  
override FlushPageOf :  
    pObj : OptimizedPersistable -> unit
```

Parameters

pObj

Type: [VelocityDb.OptimizedPersistable](#)

An object indicating what [Page](#) to flush



See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FlushUpdates Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|-------------------------------------------------------|
|  | FlushUpdates() | Write all updated data. This may free up some memory. |
|  | FlushUpdates(Database) | Write any updated/new pages for a specific Database |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.FlushUpdates Method

Write all updated data. This may free up some memory.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void FlushUpdates ()
```

VB

```
Public Overridable Sub FlushUpdates
```

C++

```
public:  
virtual void FlushUpdates ()
```

F#

```
abstract FlushUpdates : unit -> unit  
override FlushUpdates : unit -> unit
```

See Also

[SessionBase Class](#)

[FlushUpdates Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.FlushUpdates Method (Database)

Write any updated/new pages for a specific Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FlushUpdates (  
    Database db  
)
```

VB

```
Public Sub FlushUpdates (  
    db As Database  
)
```

C++

```
public:  
void FlushUpdates (  
    Database^ db  
)
```

F#

```
member FlushUpdates :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

The database for which we want to write all updated pages

See Also

[SessionBase Class](#)

[FlushUpdates Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.FlushUpdatesServers Method

Make servers Write all updated data. This may free up some memory on the servers. Only effects sessions using VelocityDbServer(s)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void FlushUpdatesServers ()
```

VB

```
Public Overridable Sub FlushUpdatesServers
```

C++

```
public:  
virtual void FlushUpdatesServers ()
```

F#

```
abstract FlushUpdatesServers : unit -> unit  
override FlushUpdatesServers : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ForceDatabaseCacheValidation Method

Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ForceDatabaseCacheValidation ()
```

VB

```
Public Overridable Sub ForceDatabaseCacheValidation
```

C++

```
public:  
virtual void ForceDatabaseCacheValidation ()
```

F#

```
abstract ForceDatabaseCacheValidation : unit -> unit  
override ForceDatabaseCacheValidation : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.GetEnumerator Method

Enumerates all open databases for this session

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Database> GetEnumerator ()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of Database)
```

C++

```
public:  
virtual IEnumerable<Database^> GetEnumerator () sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<Database>  
override GetEnumerator : unit -> IEnumerable<Database>
```

Return Value

Type: [IEnumerable\(Database\)](#)

Enumeration of open databases

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.GetVersion Method

Gets the version of [Database 0](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong GetVersion()
```

VB

```
Public Function GetVersion As ULong
```

C++

```
public:  
unsigned long long GetVersion()
```

F#

```
member GetVersion : unit -> uint64
```

Return Value

Type: [UInt64](#)

Database 0 version

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.GlobalObjWrapperGet Method

Lookup wrapper object for a non IOptimizedPersistable object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool GlobalObjWrapperGet (  
    Object lookupObj,  
    out IOptimizedPersistable pObj  
)
```

VB

```
Public Overridable Function GlobalObjWrapperGet (  
    lookupObj As Object,  
    <OutAttribute> ByRef pObj As IOptimizedPersistable  
) As Boolean
```

C++

```
public:  
virtual bool GlobalObjWrapperGet (  
    Object^ lookupObj,  
    [OutAttribute] IOptimizedPersistable^% pObj  
)
```

F#

```
abstract GlobalObjWrapperGet :  
    lookupObj : Object *  
    pObj : IOptimizedPersistable byref -> bool  
override GlobalObjWrapperGet :  
    lookupObj : Object *  
    pObj : IOptimizedPersistable byref -> bool
```

Parameters

lookupObj

Type: [System.Object](#)

object to look for

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

wrapper object

Return Value

Type: [Boolean](#)

Wrapper object

VelocityDB Class Library

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.HighestInUseDatabaseNumber Method

Figure out what the highest in use database number is and return it.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint HighestInUseDatabaseNumber ()
```

VB

```
Public Function HighestInUseDatabaseNumber As UInteger
```

C++

```
public:  
unsigned int HighestInUseDatabaseNumber ()
```

F#

```
member HighestInUseDatabaseNumber : unit -> uint32
```

Return Value

Type: [UInt32](#)

The currently largest Database Id in use reachable from this session

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.HighestInUseLocalDatabaseNumber Method

Figure out what the highest in use database number is on the local host and return it.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint HighestInUseLocalDatabaseNumber ()
```

VB

```
Public Function HighestInUseLocalDatabaseNumber As UInteger
```

C++

```
public:  
unsigned int HighestInUseLocalDatabaseNumber ()
```

F#

```
member HighestInUseLocalDatabaseNumber : unit -> uint32
```

Return Value

Type: [UInt32](#)

The currently largest Database Id in use reachable from this session located on localhost

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.IdOf Method

Finds the object id of a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong IdOf(  
    Object obj  
)
```

VB

```
Public Function IdOf (  
    obj As Object  
) As ULong
```

C++

```
public:  
unsigned long long IdOf(  
    Object^ obj  
)
```

F#

```
member IdOf :  
    obj : Object -> uint64
```

Parameters

obj

Type: [System.Object](#)

The object to find the id for

Return Value

Type: [UInt64](#)

The object id or 0 if object was not found





See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Index Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Index(T)() | Gets an index of all objects of a certain type |
|  | Index(T)(String) | Gets an index of all objects of a certain type |
|  | Index(T)(Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used). If you have updated objects persisted in a prior transaction that you want part of an index. Call FlushUpdates() or call Write()/> to add such an object to index before querying using an index./> |
|  | Index(T)(String, Database) | Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used) |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Index(*T*) Method

Gets an index of all objects of a certain type

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<T> Index<T>()
```

VB

```
Public Function Index(Of T) As BTreeSet(Of T)
```

C++

```
public:  
generic<typename T>  
BTreeSet<T>^ Index()
```

F#

```
member Index : unit -> BTreeSet<'T>
```

Type Parameters

T

The type of object indexed

Return Value

Type: [BTreeSet\(*T*\)](#)

The index

See Also

[SessionBase Class](#)

[Index Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Index(*T*) Method (String)

Gets an index of all objects of a certain type

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<T> Index<T>(
    string indexedByFieldName
)
```

VB

```
Public Function Index(Of T) (
    indexedByFieldName As String
) As BTreeSet(Of T)
```

C++

```
public:
generic<typename T>
BTreeSet<T>^ Index(
    String^ indexedByFieldName
)
```

F#

```
member Index :
    indexedByFieldName : string -> BTreeSet<'T>
```

Parameters

indexedByFieldName

Type: [System.String](#)

The field used for sorting the indexed objects

Type Parameters

T

The type of object indexed

Return Value

Type: [BTreeSet\(T\)](#)

The index

See Also

[SessionBase Class](#)

[Index Overload](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.Index(*T*) Method (Database)

Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used). If you have updated objects persisted in a prior transaction that you want part of an index. Call [FlushUpdates\(\)](#) or call [Write\(\)](#) to add such an object to index before querying using an index./>

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSetOidShort<T> Index<T>(
    Database inDatabase
)
```

VB

```
Public Function Index(Of T) (
    inDatabase As Database
) As BTreeSetOidShort(Of T)
```

C++

```
public:
generic<typename T>
BTreeSetOidShort<T>^ Index(
    Database^ inDatabase
)
```

F#

```
member Index :
    inDatabase : Database -> BTreeSetOidShort<'T>
```

Parameters

inDatabase

Type: [VelocityDb.Database](#)

Use index as created within this [Database](#)

Type Parameters

T

The type of object indexed

Return Value

Type: [BTreeSetOidShort\(T\)](#)

The index

VelocityDB Class Library

See Also

[SessionBase Class](#)

[Index Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Index(*T*) Method (String, Database)

Gets an index of all objects of a certain type within a given database (attribute OnePerDatabase must be used)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSetOidShort<T> Index<T>(
    string indexedByFieldName,
    Database inDatabase
)
```

VB

```
Public Function Index(Of T) (
    indexedByFieldName As String,
    inDatabase As Database
) As BTreeSetOidShort(Of T)
```

C++

```
public:
generic<typename T>
BTreeSetOidShort<T>^ Index(
    String^ indexedByFieldName,
    Database^ inDatabase
)
```

F#

```
member Index :
    indexedByFieldName : string *
    inDatabase : Database -> BTreeSetOidShort<'T>
```

Parameters

indexedByFieldName

Type: [System.String](#)

The field used for sorting the indexed objects

inDatabase

Type: [VelocityDb.Database](#)

Database containing index

Type Parameters

T

The type of object indexed

VelocityDB Class Library

Return Value

Type: [BTreeSetOidShort\(T\)](#)

The index

See Also

[SessionBase Class](#)

[Index Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.InUseNumberOfPages Method

Gets the number of pages currently in use by a Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ushort InUseNumberOfPages (
    Database db
)
```

VB

```
Public Overridable Function InUseNumberOfPages (
    db As Database
) As UShort
```

C++

```
public:
virtual unsigned short InUseNumberOfPages (
    Database^ db
)
```

F#

```
abstract InUseNumberOfPages :
    db : Database -> uint16
override InUseNumberOfPages :
    db : Database -> uint16
```

Parameters

db

Type: [VelocityDb.Database](#)

The Database to request info about

Return Value

Type: [UInt16](#)

The number of pages currently in use

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.IsSameHost Method

Determine if two host names represent the same tcp/ip host

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool IsSameHost(  
    string host1,  
    string host2  
)
```

VB

```
Public Shared Function IsSameHost (  
    host1 As String,  
    host2 As String  
) As Boolean
```

C++

```
public:  
static bool IsSameHost(  
    String^ host1,  
    String^ host2  
)
```

F#

```
static member IsSameHost :  
    host1 : string *  
    host2 : string -> bool
```

Parameters

host1

Type: [System.String](#)

Host name 1

host2

Type: [System.String](#)

Host name 2

Return Value

Type: [Boolean](#)

True if host is the same; otherwise false

See Also

[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.IsWorkerThread Method

Tell if code is executed by index or page write worker thread

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsWorkerThread()
```

VB

```
Public Function IsWorkerThread As Boolean
```

C++

```
public:  
bool IsWorkerThread()
```

F#

```
member IsWorkerThread : unit -> bool
```

Return Value

Type: [Boolean](#)

True, if worker thread is active thread; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.LoadFields Method

Loads all field values of an object if they are not already loaded.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void LoadFields(  
    IOptimizedPersistable pObj,  
    int depth = 2147483647  
)
```

VB

```
Public Sub LoadFields (  
    pObj As IOptimizedPersistable,  
    Optional depth As Integer = 2147483647  
)
```

C++

```
public:  
void LoadFields(  
    IOptimizedPersistable^ pObj,  
    int depth = 2147483647  
)
```

F#

```
member LoadFields :  
    pObj : IOptimizedPersistable *  
    ?depth : int  
(* Defaults:  
    let _depth = defaultArg depth 2147483647  
)  
-> unit
```

Parameters

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

Persistent object for which to load field values.

depth (Optional)

Type: [System.Int32](#)

Set this if you want to limit the depth of the graph loaded by this open.

See Also

[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.LocateDb Method

Lookup the [DatabaseLocation](#) of a [Database](#) with a specified database number.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation LocateDb(  
    uint dbNum  
)
```

VB

```
Public Function LocateDb (  
    dbNum As UInteger  
) As DatabaseLocation
```

C++

```
public:  
DatabaseLocation^ LocateDb(  
    unsigned int dbNum  
)
```

F#

```
member LocateDb :  
    dbNum : uint32 -> DatabaseLocation
```

Parameters

dbNum

Type: [System.UInt32](#)

Database id to look for

Return Value

Type: [DatabaseLocation](#)

The current [DatabaseLocation](#) containing the requested [Database](#).

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.NewDatabase Method

Create a new Database with a given database number

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Database NewDatabase(
    uint dbNum,
    uint megaBytesPresize = 0,
    string name = null,
    bool signalError = true
)
```

VB

```
Public Overridable Function NewDatabase (
    dbNum As UInteger,
    Optional megaBytesPresize As UInteger = 0,
    Optional name As String = Nothing,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ NewDatabase(
    unsigned int dbNum,
    unsigned int megaBytesPresize = 0,
    String^ name = nullptr,
    bool signalError = true
)
```

F#

```
abstract NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
    let_megaBytesPresize = defaultArg megaBytesPresize 0
    let_name = defaultArg name null
    let_signalError = defaultArg signalError true
*)
-> Database
override NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
```

VelocityDB Class Library

```
        let_megaBytesPresize = defaultArg megaBytesPresize 0
        let_name = defaultArg name null
        let_signalError = defaultArg signalError true
    *)
    -> Database
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the database to create

megaBytesPresize (Optional)

Type: [System.UInt32](#)

If you know that the Database will be large, presizing it may avoid file fragmentation. Default value is 0

name (Optional)

Type: [System.String](#)

Optionally name the new [Database](#)

signalError (Optional)

Type: [System.Boolean](#)

Optionally signal an error if creation of new [Database](#) fails

Return Value

Type: [Database](#)

The newly created [Database](#) or throws an exception if the [Database](#) already exist

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.NewLocation Method

Creates a new DatabaseLocation or updates existing ones

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual DatabaseLocation NewLocation(  
    DatabaseLocation location  
)
```

VB

```
Public Overridable Function NewLocation (  
    location As DatabaseLocation  
) As DatabaseLocation
```

C++

```
public:  
virtual DatabaseLocation^ NewLocation(  
    DatabaseLocation^ location  
)
```

F#

```
abstract NewLocation :  
    location : DatabaseLocation -> DatabaseLocation  
override NewLocation :  
    location : DatabaseLocation -> DatabaseLocation
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The input location

Return Value

Type: [DatabaseLocation](#)

The new DatabaseLocation or an existing one

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Objects(T) Method

Enumerates all the objects of a given type

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<T> Objects<T>()  
where T : OptimizedPersistable
```

VB

```
Public Function Objects(Of T As OptimizedPersistable) As IEnumerable(Of T)
```

C++

```
public:  
generic<typename T>  
where T : OptimizedPersistable  
IEnumerator<T>^ Objects()
```

F#

```
member Objects : unit -> IEnumerable<'T> when 'T : OptimizedPersistable
```

Type Parameters

T

The type of object we are looking for

Return Value

Type: [IEnumerator\(T\)](#)

An enumeration of type T

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OfType Method

Gets an object used for enumerating all objects in all Databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public OfType OfType(  
    Type type,  
    bool includeSubclasses = true,  
    bool databasePerType = true  
)
```

VB

```
Public Function OfType (  
    type As Type,  
    Optional includeSubclasses As Boolean = true,  
    Optional databasePerType As Boolean = true  
) As OfType
```

C++

```
public:  
OfType^ OfType(  
    Type^ type,  
    bool includeSubclasses = true,  
    bool databasePerType = true  
)
```

F#

```
member OfType :  
    type : Type *  
    ?includeSubclasses : bool *  
    ?databasePerType : bool  
(* Defaults:  
    let_includeSubclasses = defaultArg includeSubclasses true  
    let_databasePerType = defaultArg databasePerType true  
)  
-> OfType
```

Parameters

type

Type: [System.Type](#)

Type to look for

includeSubclasses (Optional)

Type: [System.Boolean](#)

Also return instances of sub classes or classes that implements the specified interface class

databasePerType (Optional)

Type: [System.Boolean](#)

Assume that persisted objects were made persistent the simple way using [Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#). When persisting this way, each object type gets its own [Database](#) which makes finding these objects easier and faster

Return Value

Type: [OfType](#)

The enumeration wrapper **OfType(Type, Boolean, Boolean)**object

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OidOf Method

Finds the object id of a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Oid OidOf(  
    Object obj  
)
```

VB

```
Public Function OidOf (  
    obj As Object  
) As Oid
```

C++

```
public:  
Oid OidOf(  
    Object^ obj  
)
```

F#

```
member OidOf :  
    obj : Object -> Oid
```

Parameters

obj

Type: [System.Object](#)

The object to find the id for

Return Value

Type: [Oid](#)

The object id or 0 if object was not found

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OidShortOf Method

Finds the object id of a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint OidShortOf(  
    Object obj  
)
```

VB

```
Public Function OidShortOf (  
    obj As Object  
) As UInteger
```

C++

```
public:  
unsigned int OidShortOf(  
    Object^ obj  
)
```

F#

```
member OidShortOf :  
    obj : Object -> uint32
```

Parameters

obj

Type: [System.Object](#)

The object to find the id for

Return Value

Type: [UInt32](#)

The object id or 0 if object was not found

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OpenAllDatabases Method

Open all databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overridable Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
)
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

Open for update?

Return Value

Type: [List\(Database\)](#)

List of [Database](#)

See Also

[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.OpenDatabase Method

Opens a Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Database OpenDatabase(
    uint dbNum,
    bool update = false,
    bool signalError = true
)
```

VB

```
Public Overridable Function OpenDatabase (
    dbNum As UInteger,
    Optional update As Boolean = false,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ OpenDatabase(
    unsigned int dbNum,
    bool update = false,
    bool signalError = true
)
```

F#

```
abstract OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
override OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
```


VelocityDB Class Library

Parameters

dbNum

Type: [System.UInt32](#)

The Database number of the database to open

update (Optional)

Type: [System.Boolean](#)

Open the Database for update?

signalError (Optional)

Type: [System.Boolean](#)

Signal errors if errors found when opening the Database?

Return Value

Type: [Database](#)

The opened Database

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OpenDatabaseLocations Method

Open the [DatabaseLocations](#) object and store it in the session.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocations OpenDatabaseLocations ()
```

VB

```
Public Function OpenDatabaseLocations As DatabaseLocations
```

C++

```
public:  
DatabaseLocations^ OpenDatabaseLocations ()
```

F#

```
member OpenDatabaseLocations : unit -> DatabaseLocations
```

Return Value

Type: [DatabaseLocations](#)

The [DatabaseLocations](#) used by this session

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OpenLocationDatabases Method

Opens all the databases in a given location

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual List<Database> OpenLocationDatabases (  
    DatabaseLocation location,  
    bool update = false  
)
```

VB

```
Public Overridable Function OpenLocationDatabases (  
    location As DatabaseLocation,  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenLocationDatabases (  
    DatabaseLocation^ location,  
    bool update = false  
)
```

F#

```
abstract OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The location for which to open databases

update (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Shall each database be opened for update?

Return Value

Type: [List\(Database\)](#)

A list of opened databases

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OpenPage Method

Opens a page for read, used by DatabaseManager for page browsing

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Page OpenPage(  
    Database db,  
    ushort pageNumber,  
    bool stubOnly = false  
)
```

VB

```
Public Function OpenPage (  
    db As Database,  
    pageNumber As UShort,  
    Optional stubOnly As Boolean = false  
) As Page
```

C++

```
public:  
Page^ OpenPage(  
    Database^ db,  
    unsigned short pageNumber,  
    bool stubOnly = false  
)
```

F#

```
member OpenPage :  
    db : Database *  
    pageNumber : uint16 *  
    ?stubOnly : bool  
(* Defaults:  
    let_stubOnly = defaultArg stubOnly false  
)  
-> Page
```

Parameters

db

Type: [VelocityDb.Database](#)

Database containing the requested page

pageNumber

Type: [System.UInt16](#)

Page number of the requested page

VelocityDB Class Library

stubOnly (Optional)

Type: [System.Boolean](#)

Open page but don't retrieve slot bytes? (lazy load)

Return Value

Type: [Page](#)

The requested page

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.OpenSchema Method

Get the session active schema

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Schema OpenSchema (  
    bool update  
)
```

VB

```
Public Overridable Function OpenSchema (  
    update As Boolean  
) As Schema
```

C++

```
public:  
virtual Schema^ OpenSchema (  
    bool update  
)
```

F#

```
abstract OpenSchema :  
    update : bool -> Schema  
override OpenSchema :  
    update : bool -> Schema
```

Parameters

update

Type: [System.Boolean](#)

Do update the schema?

Return Value

Type: [Schema](#)

The active schema

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ParseCsvRow Method

Parses a row in a csv file and returns an iterator of column string values

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEnumerable<string> ParseCsvRow(  
    string r  
)
```

VB

```
Public Shared Function ParseCsvRow (  
    r As String  
) As IEnumerable(Of String)
```

C++

```
public:  
static IEnumerable<String^>^ ParseCsvRow(  
    String^ r  
)
```

F#

```
static member ParseCsvRow :  
    r : string -> IEnumerable<string>
```

Parameters

r

Type: [System.String](#)

Csv row string data to be parsed

Return Value

Type: [IEnumerable\(String\)](#)

Iterator of column values







See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method

Overload List

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Persist(Object) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database. |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . |
|  Persist(Placement, IOptimizedPersistable) | Persist an object that is an OptimizedPersistable or a subclass |
|  Persist(Object, IOptimizedPersistable, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |
|  Persist(Object, Placement, Boolean) | Persist an object that may or may not be a subclass of OptimizedPersistable |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object |

See Also

[SessionBase Class](#)[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (Object)

This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Persist(  
    Object obj  
)
```

VB

```
Public Function Persist (  
    obj As Object  
) As ULong
```

C++

```
public:  
unsigned long long Persist(  
    Object^ obj  
)
```

F#

```
member Persist :  
    obj : Object -> uint64
```

Parameters

obj

Type: [System.Object](#)

The object to make persistent

Return Value

Type: [UInt64](#)

The Id of the persisted object

See Also

[SessionBase Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (IOptimizedPersistable, Nullable(UInt16))

This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides [PlacementDatabaseNumber](#) and returns something other than [DefaultPlacementDatabaseNumber](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ulong Persist (
    IOptimizedPersistable ipObj,
    ushort? objectsPerPage = null
)
```

VB

```
Public Overridable Function Persist (
    ipObj As IOptimizedPersistable,
    Optional objectsPerPage As UShort? = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    IOptimizedPersistable^ ipObj,
    Nullable<unsigned short> objectsPerPage = nullptr
)
```

F#

```
abstract Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let_objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
override Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let_objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
```

Parameters

ipObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to make persistent

VelocityDB Class Library

objectsPerPage (Optional)

Type: [System.Nullable\(UInt16\)](#)

Override of objects per page, only respected when persisting first object of some type

Return Value

Type: [UInt64](#)

The Id of the persisted object

See Also

[SessionBase Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (Placement, IOptimizedPersistable)

Persist an object that is an OptimizedPersistable or a subclass

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Persist(  
    Placement place,  
    IOptimizedPersistable obj  
)
```

VB

```
Public Function Persist (  
    place As Placement,  
    obj As IOptimizedPersistable  
) As ULong
```

C++

```
public:  
unsigned long long Persist(  
    Placement^ place,  
    IOptimizedPersistable^ obj  
)
```

F#

```
member Persist :  
    place : Placement *  
    obj : IOptimizedPersistable -> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

Provides placement guidance

obj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to persist

Return Value

Type: [UInt64](#)

The Id of the persistent object

See Also

[SessionBase Class](#)

VelocityDB Class Library

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (Object, IOptimizedPersistable, Boolean)

Persist an object that may or may not be a subclass of OptimizedPersistable

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Persist(  
    Object obj,  
    IOptimizedPersistable parent,  
    bool inFlush = false  
)
```

VB

```
Public Function Persist (  
    obj As Object,  
    parent As IOptimizedPersistable,  
    Optional inFlush As Boolean = false  
) As ULong
```

C++

```
public:  
unsigned long long Persist(  
    Object^ obj,  
    IOptimizedPersistable^ parent,  
    bool inFlush = false  
)
```

F#

```
member Persist :  
    obj : Object *  
    parent : IOptimizedPersistable *  
    ?inFlush : bool  
(* Defaults:  
    let_inFlush = defaultArg inFlush false  
)  
-> uint64
```

Parameters

obj

Type: [System.Object](#)

The object to persist.

parent

Type: [VelocityDb.IOptimizedPersistable](#)

A parent object of the object to persist.

VelocityDB Class Library

inFlush (Optional)

Type: [System.Boolean](#)

If true, disallow page flushing

Return Value

Type: [UInt64](#)

The Id of the persisted object.

See Also

[SessionBase Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (Object, Placement, Boolean)

Persist an object that may or may not be a subclass of OptimizedPersistable

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Persist(  
    Object obj,  
    Placement place,  
    bool inFlush = false  
)
```

VB

```
Public Function Persist (  
    obj As Object,  
    place As Placement,  
    Optional inFlush As Boolean = false  
) As ULong
```

C++

```
public:  
    unsigned long long Persist(  
        Object^ obj,  
        Placement^ place,  
        bool inFlush = false  
    )
```

F#

```
member Persist :  
    obj : Object *  
    place : Placement *  
    ?inFlush : bool  
(* Defaults:  
    let_inFlush = defaultArg inFlush false  
*)  
-> uint64
```

Parameters

obj

Type: [System.Object](#)

The object to persist.

place

Type: [VelocityDb.Placement](#)

A placement guidance

VelocityDB Class Library

inFlush (Optional)

Type: [System.Boolean](#)

If true, disallow page flushing

Return Value

Type: [UInt64](#)

The Id of the persisted object.

See Also

[SessionBase Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.Persist Method (Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable))

Persists an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ulong Persist(
    Placement place,
    IOptimizedPersistable pObj,
    Schema schema,
    ushort slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overridable Function Persist (
    place As Placement,
    pObj As IOptimizedPersistable,
    schema As Schema,
    Optional slotLimitPerPage As UShort = 0,
    Optional inFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    IOptimizedPersistable^ pObj,
    Schema^ schema,
    unsigned short slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
)
```

F#

```
abstract Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _slotLimitPerPage = defaultArg slotLimitPerPage 0
```

```
        let_inFlush = defaultArg inFlush false
        let_toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_slotLimitPerPage = defaultArg slotLimitPerPage 0
    let_inFlush = defaultArg inFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

Placement object determining where to persist the object

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to persist

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active session schema

slotLimitPerPage (Optional)

Type: [System.UInt16](#)

Override of how many slots per page to permit

inFlush (Optional)

Type: [System.Boolean](#)

Set to true to disallow page flushes as a side affect

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A list of objects waiting to be persisted

Return Value

Type: [UInt64](#)

Id of persisted object

See Also

[SessionBase Class](#)

[Persist Overload](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.PossiblyFlushUpdatedPages Method

Call this function if you may have updated many pages to possibly free up memory if too much memory is in use.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void PossiblyFlushUpdatedPages (  
    uint numberOfUpdatedPagesPermittedWithoutFlushCheck = 100  
)
```

VB

```
Public Overridable Sub PossiblyFlushUpdatedPages (  
    Optional numberOfUpdatedPagesPermittedWithoutFlushCheck As UInteger =  
100  
)
```

C++

```
public:  
virtual void PossiblyFlushUpdatedPages (  
    unsigned int numberOfUpdatedPagesPermittedWithoutFlushCheck = 100  
)
```

F#

```
abstract PossiblyFlushUpdatedPages :  
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32  
(* Defaults:  
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg  
numberOfUpdatedPagesPermittedWithoutFlushCheck 100  
)  
-> unit  
override PossiblyFlushUpdatedPages :  
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32  
(* Defaults:  
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg  
numberOfUpdatedPagesPermittedWithoutFlushCheck 100  
)  
-> unit
```

Parameters

numberOfUpdatedPagesPermittedWithoutFlushCheck (Optional)

Type: [System.UInt32](#)

Only do check if the number of updated pages is greater than this number

See Also

[SessionBase Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.RegisterClass Method

Register a class in the persistent schema. It is a good idea to do this explicitly instead of having it happen the first time an object of an unregistered class is made persistent

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion RegisterClass(  
    Type type  
)
```

VB

```
Public Function RegisterClass (  
    type As Type  
) As TypeVersion
```

C++

```
public:  
TypeVersion^ RegisterClass(  
    Type^ type  
)
```

F#

```
member RegisterClass :  
    type : Type -> TypeVersion
```

Parameters

type

Type: [System.Type](#)

The type to register in the persistent schema

Return Value

Type: [TypeVersion](#)

[TypeVersion](#) corresponding to type registered

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.RelocateDatabaseLocationFor Method

After moving/copying directory containing your [Database](#) with id *dbId* call this before starting a transaction to update host and directory of the associated [DatabaseLocation](#). This function does not move the database files, it only updates the [DatabaseLocation](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation RelocateDatabaseLocationFor (
    uint dbId,
    string hostName,
    string directory
)
```

VB

```
Public Function RelocateDatabaseLocationFor (
    dbId As UInteger,
    hostName As String,
    directory As String
) As DatabaseLocation
```

C++

```
public:
DatabaseLocation^ RelocateDatabaseLocationFor(
    unsigned int dbId,
    String^ hostName,
    String^ directory
)
```

F#

```
member RelocateDatabaseLocationFor :
    dbId : uint32 *
    hostName : string *
    directory : string -> DatabaseLocation
```

Parameters

dbId

Type: [System.UInt32](#)

Id of one of the [Databases](#) in location being relocated

hostName

Type: [System.String](#)

New host name

directory

VelocityDB Class Library

Type: [System.String](#)

New directory path

Return Value

Type: [DatabaseLocation](#)

Updated [DatabaseLocation](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.RelocateDefaultDatabaseLocation Method

After moving/copying directory containing your [Databases](#) 0.odb, 1.odb ... call this before starting a transaction to update host and directory of the default [DatabaseLocation](#). This function does not move the database files, it only updates the [DatabaseLocation](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DatabaseLocation RelocateDefaultDatabaseLocation ()
```

VB

```
Public Function RelocateDefaultDatabaseLocation As DatabaseLocation
```

C++

```
public:  
DatabaseLocation^ RelocateDefaultDatabaseLocation ()
```

F#

```
member RelocateDefaultDatabaseLocation : unit -> DatabaseLocation
```

Return Value

Type: [DatabaseLocation](#)

The default [DatabaseLocation](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ReplacePage Method

Only to be used in VelocityDbExtensions while syncing databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void ReplacePage(  
    Database databaseToUpdate,  
    Page pageToUpdate,  
    Page pageToRead  
)
```

VB

```
Public Overridable Sub ReplacePage (  
    databaseToUpdate As Database,  
    pageToUpdate As Page,  
    pageToRead As Page  
)
```

C++

```
public:  
virtual void ReplacePage(  
    Database^ databaseToUpdate,  
    Page^ pageToUpdate,  
    Page^ pageToRead  
)
```

F#

```
abstract ReplacePage :  
    databaseToUpdate : Database *  
    pageToUpdate : Page *  
    pageToRead : Page -> unit  
override ReplacePage :  
    databaseToUpdate : Database *  
    pageToUpdate : Page *  
    pageToRead : Page -> unit
```

Parameters

databaseToUpdate

Type: [VelocityDb.Database](#)

Database being synced

pageToUpdate

Type: [VelocityDb.Page](#)

Page to sync

VelocityDB Class Library

pageToRead

Type: [VelocityDb.Page](#)

Page to sync from



See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ReplacePersistedType Method

Overload List

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReplacePersistedType(String, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old Type loaded. Be sure to backup all your data before doing this. |
|  ReplacePersistedType(Type, Type) | Updates the Type associated with a VelocityDbType . This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior Type loaded as it is defined in the current Schema for this to work. Once the update has been done, you should no longer need to keep the old Type loaded. Be sure to backup all your data before doing this. |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.ReplacePersistedType Method (String, Type)

Updates the [Type](#) associated with a [VelocityDbType](#). This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior [Type](#) loaded as it is defined in the current [Schema](#) for this to work. Once the update has been done, you should no longer need to keep the old [Type](#) loaded. Be sure to backup all your data before doing this.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ReplacePersistedType (
    string oldTypeFullAssemblyName,
    Type newType
)
```

VB

```
Public Function ReplacePersistedType (
    oldTypeFullAssemblyName As String,
    newType As Type
) As Boolean
```

C++

```
public:
bool ReplacePersistedType (
    String^ oldTypeFullAssemblyName,
    Type^ newType
)
```

F#

```
member ReplacePersistedType :
    oldTypeFullAssemblyName : string *
    newType : Type -> bool
```

Parameters

oldTypeFullAssemblyName

Type: [System.String](#)

The assembly qualified name of the [Type](#) currently used within the [Schema](#)

newType

Type: [System.Type](#)

The [Type](#) you want to replace with in the [Schema](#). This new [Type](#) must NOT already be part of the schema.

VelocityDB Class Library

Return Value

Type: [Boolean](#)

true if old type name was found; otherwise false

See Also

[SessionBase Class](#)

[ReplacePersistedType Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.ReplacePersistedType Method (Type, Type)

Updates the [Type](#) associated with a [VelocityDbType](#). This can be used when you want to rename a class or move a class to a different namespace. You will need to have the prior [Type](#) loaded as it is defined in the current [Schema](#) for this to work. Once the update has been done, you should no longer need to keep the old [Type](#) loaded. Be sure to backup all your data before doing this.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ReplacePersistedType (  
    Type oldType,  
    Type newType  
)
```

VB

```
Public Sub ReplacePersistedType (  
    oldType As Type,  
    newType As Type  
)
```

C++

```
public:  
void ReplacePersistedType (  
    Type^ oldType,  
    Type^ newType  
)
```

F#

```
member ReplacePersistedType :  
    oldType : Type *  
    newType : Type -> unit
```

Parameters

oldType

Type: [System.Type](#)

The [Type](#) currently used within the [Schema](#)

newType

Type: [System.Type](#)

The [Type](#) you want to replace with in the [Schema](#). This new [Type](#) must NOT already be part of the schema.

VelocityDB Class Library

See Also



[SessionBase Class](#)

[ReplacePersistedType Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.RestoreFrom Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | RestoreFrom(DatabaseLocation, DateTime) | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. |
|  | RestoreFrom(String, String, UInt32, DateTime) | Restores a DatabaseLocation |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.RestoreFrom Method (DatabaseLocation, DateTime)

Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RestoreFrom(  
    DatabaseLocation backupLocation,  
    DateTime upToTime  
)
```

VB

```
Public Overridable Sub RestoreFrom (  
    backupLocation As DatabaseLocation,  
    upToTime As DateTime  
)
```

C++

```
public:  
virtual void RestoreFrom(  
    DatabaseLocation^ backupLocation,  
    DateTime upToTime  
)
```

F#

```
abstract RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit  
override RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit
```

Parameters

backupLocation

Type: [VelocityDb.DatabaseLocation](#)

The location to restore from

upToTime

Type: [System.DateTime](#)

Restore location up to a given DateTime. Data backed up after this time will not be restored.

See Also

[SessionBase Class](#)

[RestoreFrom Overload](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionBase.RestoreFrom Method (String, String, UInt32, DateTime)

Restores a [DatabaseLocation](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RestoreFrom(  
    string host,  
    string path,  
    uint backupDbNum,  
    DateTime upToTime  
)
```

VB

```
Public Overridable Sub RestoreFrom (  
    host As String,  
    path As String,  
    backupDbNum As UInteger,  
    upToTime As DateTime  
)
```

C++

```
public:  
virtual void RestoreFrom(  
    String^ host,  
    String^ path,  
    unsigned int backupDbNum,  
    DateTime upToTime  
)
```

F#

```
abstract RestoreFrom :  
    host : string *  
    path : string *  
    backupDbNum : uint32 *  
    upToTime : DateTime -> unit  
override RestoreFrom :  
    host : string *  
    path : string *  
    backupDbNum : uint32 *  
    upToTime : DateTime -> unit
```

Parameters

host

Type: [System.String](#)

Host name of backup

VelocityDB Class Library

path

Type: [System.String](#)

Directory path to backup

backupDbNum

Type: [System.UInt32](#)

Initial database number used by backup [DatabaseLocation](#)

upToTime

Type: [System.DateTime](#)

Restore data up to this time

See Also

[SessionBase Class](#)

[RestoreFrom Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.SetMinMaxStringIntern Method

A persistent string read into memory from a [Database](#) is interned using [Intern\(String\)](#) to speed up comparisons and conserve memory (when the same string occurs in many places)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SetMinMaxStringIntern(  
    ushort minSize,  
    ushort maxSize  
)
```

VB

```
Public Shared Sub SetMinMaxStringIntern (  
    minSize As UShort,  
    maxSize As UShort  
)
```

C++

```
public:  
static void SetMinMaxStringIntern(  
    unsigned short minSize,  
    unsigned short maxSize  
)
```

F#

```
static member SetMinMaxStringIntern :  
    minSize : uint16 *  
    maxSize : uint16 -> unit
```

Parameters

minSize

Type: [System.UInt16](#)

Selected minimum plus one string length of a string to be interned

maxSize

Type: [System.UInt16](#)

Selected maximum string length for a string to be interned

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SetTraceAllDbActivity Method

Enable some Trace output about persistent events related to all databases and pages. Add a Trace listener to get to Console window: `Trace.Listeners.Add(new ConsoleTraceListener());`

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetTraceAllDbActivity(  
    bool enable = true  
)
```

VB

```
Public Sub SetTraceAllDbActivity (  
    Optional enable As Boolean = true  
)
```

C++

```
public:  
void SetTraceAllDbActivity(  
    bool enable = true  
)
```

F#

```
member SetTraceAllDbActivity :  
    ?enable : bool  
(* Defaults:  
    let _enable = defaultArg enable true  
*)  
-> unit
```

Parameters

enable (Optional)

Type: [System.Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SetTraceDbActivity Method

Enable some Trace output about persistent events related to a Database. Add a Trace listener to get to Console window: `Trace.Listeners.Add(new ConsoleTraceListener());`

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetTraceDbActivity(  
    uint dbId  
)
```

VB

```
Public Sub SetTraceDbActivity (  
    dbId As UInteger  
)
```

C++

```
public:  
void SetTraceDbActivity(  
    unsigned int dbId  
)
```

F#

```
member SetTraceDbActivity :  
    dbId : uint32 -> unit
```

Parameters

dbId

Type: [System.UInt32](#)

The Database number of the Database to enable persistent trace information for.

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.StringToType Method

Gets a [Type](#) given a string representation of a type

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Type StringToType(  
    string typeAsString  
)
```

VB

```
Public Shared Function StringToType (  
    typeAsString As String  
) As Type
```

C++

```
public:  
static Type^ StringToType(  
    String^ typeAsString  
)
```

F#

```
static member StringToType :  
    typeAsString : string -> Type
```

Parameters

typeAsString

Type: [System.String](#)

The string representation of a [Type](#)

Return Value

Type: [Type](#)

Type from string

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SubscribeToChanges Method

Subscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void SubscribeToChanges (
    Type aType,
    string notifyIfTrueProperty = null
)
```

VB

```
Public Overridable Sub SubscribeToChanges (
    aType As Type,
    Optional notifyIfTrueProperty As String = Nothing
)
```

C++

```
public:
virtual void SubscribeToChanges (
    Type^ aType,
    String^ notifyIfTrueProperty = nullptr
)
```

F#

```
abstract SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
override SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are interested in being notified about when persistent instances changes within a database.

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a Boolean property. When property returns true, a change notification is send otherwise no notification is send. Leave as null to get a notification any time any instance of the selected type changes

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.TryDatabase Method

Gets the current database to try

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Database TryDatabase(  
    Placement place  
)
```

VB

```
Public Function TryDatabase (  
    place As Placement  
) As Database
```

C++

```
public:  
Database^ TryDatabase(  
    Placement^ place  
)
```

F#

```
member TryDatabase :  
    place : Placement -> Database
```

Parameters

place

Type: [VelocityDb.Placement](#)

the placement object to retrieve from

Return Value

Type: [Database](#)

A database or null if not cached

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Unpersist Method

Use this when you want to delete objects that are not assignable as [IOptimizedPersistable](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public virtual bool Unpersist(  
    Object obj  
)
```

```
VB  
Public Overridable Function Unpersist (  
    obj As Object  
) As Boolean
```

```
C++  
public:  
virtual bool Unpersist(  
    Object^ obj  
)
```

```
F#  
abstract Unpersist :  
    obj : Object -> bool  
override Unpersist :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to be deleted from persistent storage

Return Value

Type: [Boolean](#)

true if object found and deleted; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UnsubscribeToChanges Method

Unsubscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void UnsubscribeToChanges (
    Type aType,
    string notifyIfTrueProperty = null
)
```

VB

```
Public Overridable Sub UnsubscribeToChanges (
    aType As Type,
    Optional notifyIfTrueProperty As String = Nothing
)
```

C++

```
public:
virtual void UnsubscribeToChanges (
    Type^ aType,
    String^ notifyIfTrueProperty = nullptr
)
```

F#

```
abstract UnsubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
override UnsubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are no longer interested in being notified about when persistent instances changes within a database.

VelocityDB Class Library

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a Boolean property.

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateClass Method

Let VelocityDB know that changes have been made to a class so that the schema manager will create a new [TypeVersion](#) and use the new version for all new objects of the specified Type.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion UpdateClass(  
    Type type  
)
```

VB

```
Public Function UpdateClass (  
    type As Type  
) As TypeVersion
```

C++

```
public:  
TypeVersion^ UpdateClass(  
    Type^ type  
)
```

F#

```
member UpdateClass :  
    type : Type -> TypeVersion
```

Parameters

type

Type: [System.Type](#)

The type that has been updated (new/removed/modified fields)

Return Value

Type: [TypeVersion](#)

The new Type description

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateDatabase Method

Request an update lock on a database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public virtual bool UpdateDatabase(  
    Database db  
)
```

```
VB  
Public Overridable Function UpdateDatabase (  
    db As Database  
) As Boolean
```

```
C++  
public:  
virtual bool UpdateDatabase(  
    Database^ db  
)
```

```
F#  
abstract UpdateDatabase :  
    db : Database -> bool  
override UpdateDatabase :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to update

Return Value

Type: [Boolean](#)

true if Database was updated, otherwise throws an exception

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method

Overload List

| Name | Description |
|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| ⇒ UpdateObject(Object) | Tag an object as updated so that it will be updated persistently |
| ⇒ UpdateObject(IOptimizedPersistable) | Tag an object as updated so that it will be updated persistently |
| ⇒ UpdateObject(Object, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
| ⇒ UpdateObject(IOptimizedPersistable, Action) | Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done. |
| ⇒ UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method (Object)

Tag an object as updated so that it will be updated persistently

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool UpdateObject (  
    Object obj  
)
```

VB

```
Public Overridable Function UpdateObject (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject (  
    Object^ obj  
)
```

F#

```
abstract UpdateObject :  
    obj : Object -> bool  
override UpdateObject :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to update.

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

See Also

[SessionBase Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method (IOptimizedPersistable)

Tag an object as updated so that it will be updated persistently

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool UpdateObject(  
    IOptimizedPersistable obj  
)
```

VB

```
Public Overridable Function UpdateObject (  
    obj As IOptimizedPersistable  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject(  
    IOptimizedPersistable^ obj  
)
```

F#

```
abstract UpdateObject :  
    obj : IOptimizedPersistable -> bool  
override UpdateObject :  
    obj : IOptimizedPersistable -> bool
```

Parameters

obj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to update.

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

See Also

[SessionBase Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method (Object, Action)

Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool UpdateObject(  
    Object obj,  
    Action codeBlock  
)
```

VB

```
Public Overridable Function UpdateObject (  
    obj As Object,  
    codeBlock As Action  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject(  
    Object^ obj,  
    Action^ codeBlock  
)
```

F#

```
abstract UpdateObject :  
    obj : Object *  
    codeBlock : Action -> bool  
override UpdateObject :  
    obj : Object *  
    codeBlock : Action -> bool
```

Parameters

obj

Type: [System.Object](#)

Object to update

codeBlock

Type: [System.Action](#)

Block of code doing actual update to the object

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

VelocityDB Class Library

See Also

[SessionBase Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method (IOptimizedPersistable, Action)

Provides a way to update an object in a multi threaded scenario where we must make sure that object page isn't flushed until updates to object are done.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool UpdateObject(  
    IOptimizedPersistable obj,  
    Action codeBlock  
)
```

VB

```
Public Overridable Function UpdateObject (  
    obj As IOptimizedPersistable,  
    codeBlock As Action  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject(  
    IOptimizedPersistable^ obj,  
    Action^ codeBlock  
)
```

F#

```
abstract UpdateObject :  
    obj : IOptimizedPersistable *  
    codeBlock : Action -> bool  
override UpdateObject :  
    obj : IOptimizedPersistable *  
    codeBlock : Action -> bool
```

Parameters

obj

Type: [VelocityDb.IOptimizedPersistable](#)

Object to update

codeBlock

Type: [System.Action](#)

Block of code doing actual update to the object

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

VelocityDB Class Library

See Also

[SessionBase Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdateObject Method (IOptimizedPersistable, Boolean, Boolean)

Updates an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool UpdateObject(
    IOptimizedPersistable obj,
    bool inFlush,
    bool deleteObjFromIndexes = true
)
```

VB

```
Public Overridable Function UpdateObject (
    obj As IOptimizedPersistable,
    inFlush As Boolean,
    Optional deleteObjFromIndexes As Boolean = true
) As Boolean
```

C++

```
public:
virtual bool UpdateObject(
    IOptimizedPersistable^ obj,
    bool inFlush,
    bool deleteObjFromIndexes = true
)
```

F#

```
abstract UpdateObject :
    obj : IOptimizedPersistable *
    inFlush : bool *
    ?deleteObjFromIndexes : bool
(* Defaults:
    let _deleteObjFromIndexes = defaultArg deleteObjFromIndexes true
*)
-> bool
override UpdateObject :
    obj : IOptimizedPersistable *
    inFlush : bool *
    ?deleteObjFromIndexes : bool
(* Defaults:
    let _deleteObjFromIndexes = defaultArg deleteObjFromIndexes true
*)
-> bool
```

VelocityDB Class Library

Parameters

obj

Type: [VelocityDb.IOptimizedPersistable](#)

Object to be updated

inFlush

Type: [System.Boolean](#)

Are we doing this as part of a page flush? Set to true to avoid a recursive page flush

deleteObjFromIndexes (Optional)

Type: [System.Boolean](#)

Set to false if you know that this object is not part of any index. Safe to leave set to true in any case

Return Value

Type: [Boolean](#)

True if object is updated; otherwise false

See Also

[SessionBase Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionBase.UpdatePage Method

Update a [Page](#). Used by VelocityDBExtensions project

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool UpdatePage(  
    ref Page page  
)
```

VB

```
Public Function UpdatePage (  
    ByRef page As Page  
) As Boolean
```

C++

```
public:  
bool UpdatePage (  
    Page^% page  
)
```

F#

```
member UpdatePage :  
    page : Page byref -> bool
```

Parameters

page

Type: [VelocityDb.Page](#)

[Page](#)

to update

Return Value

Type: [Boolean](#)

true if updated OK; otherwise false

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Verify Method

Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Verify()
```

VB

```
Public Overridable Sub Verify
```

C++

```
public:  
virtual void Verify()
```

F#

```
abstract Verify : unit -> unit  
override Verify : unit -> unit
```

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.WaitForIndexUpdates Method

Internal Use and in VelocityDBExtensions

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public void WaitForIndexUpdates (  
    bool throwIfError = true  
)
```

```
VB  
Public Sub WaitForIndexUpdates (  
    Optional throwIfError As Boolean = true  
)
```

```
C++  
public:  
void WaitForIndexUpdates (  
    bool throwIfError = true  
)
```

```
F#  
member WaitForIndexUpdates :  
    ?throwIfError : bool  
(* Defaults:  
    let_throwIfError = defaultArg throwIfError true  
*)  
-> unit
```

Parameters

throwIfError (Optional)

Type: [System.Boolean](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.WritePageBytes Method

[Missing <summary> documentation for

"M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void WritePageBytes (
    byte[] bytes,
    Page page,
    Dictionary<ushort, long> dbPageWrites,
    ulong transaction,
    Stream fStream,
    byte[] pageInfoBytes
)
```

VB

```
Public Overridable Sub WritePageBytes (
    bytes As Byte(),
    page As Page,
    dbPageWrites As Dictionary(Of UShort, Long),
    transaction As ULong,
    fStream As Stream,
    pageInfoBytes As Byte()
)
```

C++

```
public:
virtual void WritePageBytes (
    array<unsigned char>^ bytes,
    Page^ page,
    Dictionary<unsigned short, long long>^ dbPageWrites,
    unsigned long long transaction,
    Stream^ fStream,
    array<unsigned char>^ pageInfoBytes
)
```

F#

```
abstract WritePageBytes :
    bytes : byte[] *
    page : Page *
    dbPageWrites : Dictionary<uint16, int64> *
    transaction : uint64 *
    fStream : Stream *
    pageInfoBytes : byte[] -> unit
```



```
override WritePageBytes :  
    bytes : byte[] *  
    page : Page *  
    dbPageWrites : Dictionary<uint16, int64> *  
    transaction : uint64 *  
    fStream : Stream *  
    pageInfoBytes : byte[] -> unit
```

Parameters

bytes

Type: [System.Byte\[\]](#)

[Missing <param name="bytes"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

dbPageWrites

Type: [System.Collections.Generic.Dictionary{UInt16, Int64}](#)

[Missing <param name="dbPageWrites"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

transaction

Type: [System.UInt64](#)

[Missing <param name="transaction"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

fStream

Type: [System.IO.Stream](#)

[Missing <param name="fStream"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

pageInfoBytes

Type: [System.Byte\[\]](#)

[Missing <param name="pageInfoBytes"/> documentation for "M:VelocityDb.Session.SessionBase.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collectio

```
ns.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[]]"  
]
```

See Also













[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.SessionBase Fields

The [SessionBase](#) type exposes the following members.

Fields

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Assembly | |
|   | IndexDescriptorDatabaseNumber | Database 3 is dedicated to keeping track of index usage |
|   | s_serverTcpIpPortNumber | By default port number 7031 (7032 if .net core) is used for communication with VelocityDBServer. Change the value here and also in VelocityDbServer.exe.config if you need to use a different port number. |
|   | s_typeDatabaseIdOffsetFromTypeShortId | Determines database Id to use for a certain VelocityDbType instance |
|   | s_typeDatabaseIdOverflowIndexCollisionOffset | If additional databases are required for storing instances of some index, start by attempting creating database at this offset from original |
|   | s_typeDatabaseIdOverflowOffset | If additional databases are required for storing instances of some type, start by attempting creating database at this offset from original |

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Assembly Field

[Missing <summary> documentation for "F:VelocityDb.Session.SessionBase.Assembly"]

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Assembly Assembly
```

VB

```
Public Shared ReadOnly Assembly As Assembly
```

C++

```
public:  
static initonly Assembly^ Assembly
```

F#

```
static val Assembly: Assembly
```

Field Value

Type: [Assembly](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.IndexDescriptorDatabaseNumber Field

Database 3 is dedicated to keeping track of index usage

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint IndexDescriptorDatabaseNumber = 3
```

VB

```
Public Const IndexDescriptorDatabaseNumber As UInteger = 3
```

C++

```
public:  
literal unsigned int IndexDescriptorDatabaseNumber = 3
```

F#

```
static val mutable IndexDescriptorDatabaseNumber: uint32
```

Field Value

Type: [UInt32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.s_serverTcpIpPortNumber Field

By default port number 7031 (7032 if .net core) is used for communication with VelocityDBServer. Change the value here and also in VelocityDbServer.exe.config if you need to use a different port number.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int s_serverTcpIpPortNumber
```

VB

```
Public Shared s_serverTcpIpPortNumber As Integer
```

C++

```
public:  
static int s_serverTcpIpPortNumber
```

F#

```
static val mutable s_serverTcpIpPortNumber: int
```

Field Value

Type: [Int32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.s_typeDatabaseIdOffsetFromTypeShortId Field

Determines database Id to use for a certain VelocityDbType instance

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly int s_typeDatabaseIdOffsetFromTypeShortId
```

VB

```
Public Shared ReadOnly s_typeDatabaseIdOffsetFromTypeShortId As Integer
```

C++

```
public:  
static initonly int s_typeDatabaseIdOffsetFromTypeShortId
```

F#

```
static val s_typeDatabaseIdOffsetFromTypeShortId: int
```

Field Value

Type: [Int32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.s_typeDatabaseIdOverflowIndexCollisionOffset Field

If additional databases are required for storing instances of some index, start by attempting creating database at this offset from original

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly uint s_typeDatabaseIdOverflowIndexCollisionOffset
```

VB

```
Public Shared ReadOnly s_typeDatabaseIdOverflowIndexCollisionOffset As  
UInteger
```

C++

```
public:  
static initonly unsigned int s_typeDatabaseIdOverflowIndexCollisionOffset
```

F#

```
static val s_typeDatabaseIdOverflowIndexCollisionOffset: uint32
```

Field Value

Type: [UInt32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.s_typeDatabaseIdOverflowOffset Field

If additional databases are required for storing instances of some type, start by attempting creating database at this offset from original

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly uint s_typeDatabaseIdOverflowOffset
```

VB

```
Public Shared ReadOnly s_typeDatabaseIdOverflowOffset As UInteger
```

C++

```
public:  
static initonly unsigned int s_typeDatabaseIdOverflowOffset
```

F#

```
static val s_typeDatabaseIdOverflowOffset: uint32
```

Field Value

Type: [UInt32](#)

See Also

[SessionBase Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Transaction Class

Helper object to enable automatic rollback/abort of transaction if transaction isn't committed within it's scope.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Session.SessionBase.Transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Transaction : IDisposable
```

VB

```
Public Class Transaction  
    Implements IDisposable
```

C++




```
public ref class Transaction : IDisposable
```

F#

```
type Transaction =  
    class  
        interface IDisposable  
    end
```

The SessionBase.Transaction type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------|
|  | Commit | Commits the active transaction |
|  | Dispose | If in update transaction; rollback (abort) transaction. If in read only transaction; commit transaction. |
|  | Rollback | Aborts the current transaction and rolls it back. Same as Abort() |




See Also

[VelocityDb.Session Namespace](#)

Transaction.Transaction Methods

The [SessionBase.Transaction](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------|
|  | Commit | Commits the active transaction |
|  | Dispose | If in update transaction; rollback (abort) transaction. If in read only transaction; commit transaction. |
|  | Rollback | Aborts the current transaction and rolls it back. Same as Abort() |

See Also

[SessionBase.Transaction Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Transaction.Commit Method

Commits the active transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
void Commit ()
```

F#

```
member Commit : unit -> unit
```

See Also

[SessionBase.Transaction Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Transaction.Dispose Method

If in update transaction; rollback (abort) transaction. If in read only transaction; commit transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[SessionBase.Transaction Class](#)

[VelocityDb.Session Namespace](#)

SessionBase.Transaction.Rollback Method

Aborts the current transaction and rolls it back. Same as [Abort\(\)](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
void Rollback()
```

F#

```
member Rollback : unit -> unit
```

See Also

[SessionBase.Transaction Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServer Class

Use this session class when [Database](#) level locking is acceptable and [Databases](#) are local or reachable by UNC paths.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

VelocityDb.Session.SessionNoServer

[VelocityDb.Session.SessionNoServerShared](#)

[VelocityDBExtensions2.AzureBlobSession](#)

[VelocityDBExtensions2.AzureSession](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class SessionNoServer : SessionBase,
    IDisposable
```

VB

```
Public Class SessionNoServer
    Inherits SessionBase
    Implements IDisposable
```

C++


```
public ref class SessionNoServer : public SessionBase,
    IDisposable
```

F#


```
type SessionNoServer =
    class
        inherit SessionBase
        interface IDisposable
    end
```

The **SessionNoServer** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------|
|  | SessionNoServer | Creates a new session. For a thread safe version of this class, use SessionNoServerShared |










Properties

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AssumeLocalHost | SessionNoServer ignores DatabaseLocation HostName since it cannot connect to a VelocityDbServer. (Overrides SessionBase.AssumeLocalHost .) |

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
|  UpdateDatabase | Request an update lock on a database (Overrides SessionBase.UpdateDatabase(Database) .) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv .) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport .) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport .) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport .) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv .) |
|  ImportJson(T) | (Defined by JsonImportExport .) |
|  MicrosoftSync | (Defined by Sync .) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync .) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync .) |

See Also

[VelocityDb.Session Namespace](#)

SessionNoServer Constructor

Creates a new session. For a thread safe version of this class, use [SessionNoServerShared](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionNoServer(
    string systemDir,
    int waitForLockMilliseconds = 5000,
    bool optimisticLocking = true,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes,
    bool enableDatabaseCache = true
)
```

VB

```
Public Sub New (
    systemDir As String,
    Optional waitForLockMilliseconds As Integer = 5000,
    Optional optimisticLocking As Boolean = true,
    Optional enablePageCache As Boolean = true,
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes,
    Optional enableDatabaseCache As Boolean = true
)
```

C++

```
public:
    SessionNoServer(
        String^ systemDir,
        int waitForLockMilliseconds = 5000,
        bool optimisticLocking = true,
        bool enablePageCache = true,
        CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes,
        bool enableDatabaseCache = true
    )
```

F#

```
new :
    systemDir : string *
    ?waitForLockMilliseconds : int *
    ?optimisticLocking : bool *
    ?enablePageCache : bool *
    ?objectCachingDefaultPolicy : CacheEnum *
    ?enableDatabaseCache : bool
(* Defaults:
    let _waitForLockMilliseconds = defaultArg waitForLockMilliseconds 5000
    let _optimisticLocking = defaultArg optimisticLocking true
    let _enablePageCache = defaultArg enablePageCache true
```

```
        let_objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
        let_enableDatabaseCache = defaultArg enableDatabaseCache true
*)
-> SessionNoServer
```

Parameters

systemDir

Type: [System.String](#)

The startup location directory path. Path can be an absolute (full) path or a path relative to **s_baseDatabasePath**

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

The desired maximum lock wait time

optimisticLocking (Optional)

Type: [System.Boolean](#)

Use optimistic locking. With optimistic locking, readers are always permitted and multiple updaters are permitted to update the same Databases/Pages but only the first transaction to commit or flush an updated Database/Page will be successful in making the committed change. Other updaters will get an `OptimisticLockingException` if they try to commit/flush updates to the same Pages/Database. See http://en.wikipedia.org/wiki/Optimistic_concurrency_control for further explanation of optimistic locking. If optimistic locking is not enabled then pessimistic locking is used. See: http://en.wikipedia.org/wiki/Concurrency_control

enablePageCache (Optional)

Type: [System.Boolean](#)

If strong reference page caching isn't desired, turn it off by setting to false

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

Determines default value of [Cache](#)

enableDatabaseCache (Optional)

Type: [System.Boolean](#)

Is this session going to enable strong reference [Database](#) caching?

See Also


[SessionNoServer Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServer.SessionNoServer Properties

The [SessionNoServer](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AssumeLocalHost | SessionNoServer ignores DatabaseLocation HostName since it cannot connect to a VelocityDbServer. (Overrides SessionBase.AssumeLocalHost .) |

See Also

[SessionNoServer Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServer.AssumeLocalHost Property

SessionNoServer ignores DatabaseLocation HostName since it cannot connect to a VelocityDbServer.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AssumeLocalHost { get; }
```

VB

```
Public Overrides ReadOnly Property AssumeLocalHost As Boolean  
    Get
```

C++

```
public:  
virtual property bool AssumeLocalHost {  
    bool get () override;  
}
```

F#

```
abstract AssumeLocalHost : bool with get  
override AssumeLocalHost : bool with get
```

Property Value

Type: [Boolean](#)

See Also


[SessionNoServer Class](#)

[VelocityDb.Session Namespace](#)










SessionNoServer.SessionNoServer Methods

The [SessionNoServer](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
|  UpdateDatabase | Request an update lock on a database (Overrides SessionBase.UpdateDatabase(Database.)) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[SessionNoServer Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServer.UpdateDatabase Method

Request an update lock on a database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateDatabase(  
    Database db  
)
```

VB

```
Public Overrides Function UpdateDatabase (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool UpdateDatabase(  
    Database^ db  
) override
```

F#

```
abstract UpdateDatabase :  
    db : Database -> bool  
override UpdateDatabase :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to update

Return Value

Type: [Boolean](#)

true if Database was updated, otherwise throws an exception

See Also

[SessionNoServer Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared Class

Same as SessionNoServer but SessionNoServerShared is thread safe so it can be shared between threads.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

[VelocityDb.Session.SessionNoServer](#)

VelocityDb.Session.SessionNoServerShared

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class SessionNoServerShared : SessionNoServer,
    IDisposable
```

VB

```
Public Class SessionNoServerShared
    Inherits SessionNoServer
    Implements IDisposable
```

C++


```
public ref class SessionNoServerShared : public SessionNoServer,
    IDisposable
```

F#




```
type SessionNoServerShared =
    class
        inherit SessionNoServer
        interface IDisposable
    end
```

The **SessionNoServerShared** type exposes the following members.











Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------|
|  | SessionNoServerShared | Creates a new session. For a thread safe version of this class, use SessionNoServerShared |














Properties




| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. (Overrides SessionBase.AddToIndexInSeperateThread.) |
|  Databases | Gets a list of the currently opened databases (Overrides SessionBase.Databases.) |
|  WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. (Overrides SessionBase.WriteToDiskInSeperateDatabaseThreads.) |

Methods










| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Transaction control, abort the current transaction (Overrides SessionBase.Abort().) |
|  BeginRead | Transaction control, begin a read only transaction (Overrides SessionBase.BeginRead(Boolean).) |
|  BeginUpdate | Transaction control, begin an update transaction. (Overrides SessionBase.BeginUpdate(Boolean).) |
|  Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() (Overrides SessionBase.Checkpoint().) |
|  ClearCache | Clears page cache and closes databases. Avoid using this one for now. (Overrides SessionBase.ClearCache().) |
|  ClearCachedPages | Clear cache of cached pages (Overrides SessionBase.ClearCachedPages().) |
|  ClearPageCache | Clears cached pages from cache including page weak references. (Overrides SessionBase.ClearPageCache().) |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact().) |
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database).) |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides SessionBase.CopyAllDatabasesTo(String, Boolean).) |

| | | |
|---|--------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ⇒ | CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened. (Overrides SessionBase.CrossTransactionCache(Database, Boolean).) |
| ⇒ | DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides SessionBase.DeleteLocation(DatabaseLocation, Boolean).) |
| ⇒ | FileOpen | Opens a Database file for read/update. Internal use only. (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean).) |
| ⇒ | FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written. (Overrides SessionBase.FlushPageOf(OptimizedPersistable).) |
| ⇒ | FlushUpdates | Write all updated data. This may free up some memory. (Overrides SessionBase.FlushUpdates().) |
| ⇒ | ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. (Overrides SessionBase.ForceDatabaseCacheValidation().) |
| ⇒ | GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object (Overrides SessionBase.GlobalObjWrapperGet(Object, IOptimizedPersistable).) |
| ⇒ | NewDatabase | Create a new Database with a given database number (Overrides SessionBase.NewDatabase(UInt32, UInt32, String, Boolean).) |
| ⇒ | NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides SessionBase.NewLocation(DatabaseLocation).) |
| ⇒ | Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
| ⇒ | Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |
|  OpenAllDatabases | Open all databases (Overrides SessionBase.OpenAllDatabases(Boolean).) |
|  OpenDatabase | Opens a Database (Overrides SessionBase.OpenDatabase(UInt32, Boolean, Boolean).) |
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides SessionBase.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  OpenSchema | Get the session active schema (Overrides SessionBase.OpenSchema(Boolean).) |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)).) |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)).) |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. (Overrides SessionBase.PossiblyFlushUpdatedPages(UInt32).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides SessionBase.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.SubscribeToChanges(Type, String).) |
|  Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable (Overrides SessionBase.Unpersist(Object).) |
|  UpdateDatabase | Request an update lock on a database (Overrides SessionNoServer.UpdateDatabase(Database).) |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |
|  Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. (Overrides SessionBase.Verify().) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[VelocityDb.Session Namespace](#)

SessionNoServerShared Constructor

Creates a new session. For a thread safe version of this class, use [SessionNoServerShared](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionNoServerShared(
    string systemDir,
    int waitForLockMilliseconds = 5000,
    bool optimisticLocking = true,
    bool enablePageCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes,
    bool enableDatabaseCache = true
)
```

VB

```
Public Sub New (
    systemDir As String,
    Optional waitForLockMilliseconds As Integer = 5000,
    Optional optimisticLocking As Boolean = true,
    Optional enablePageCache As Boolean = true,
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes,
    Optional enableDatabaseCache As Boolean = true
)
```

C++

```
public:
    SessionNoServerShared(
        String^ systemDir,
        int waitForLockMilliseconds = 5000,
        bool optimisticLocking = true,
        bool enablePageCache = true,
        CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes,
        bool enableDatabaseCache = true
    )
```

F#

```
new :
    systemDir : string *
    ?waitForLockMilliseconds : int *
    ?optimisticLocking : bool *
    ?enablePageCache : bool *
    ?objectCachingDefaultPolicy : CacheEnum *
    ?enableDatabaseCache : bool
(* Defaults:
    let _waitForLockMilliseconds = defaultArg waitForLockMilliseconds 5000
    let _optimisticLocking = defaultArg optimisticLocking true
    let _enablePageCache = defaultArg enablePageCache true
```

```
        let_objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
        let_enableDatabaseCache = defaultArg enableDatabaseCache true
*)
-> SessionNoServerShared
```

Parameters

systemDir

Type: [System.String](#)

The startup location directory path. Path can be an absolute (full) path or a path relative to **s_baseDatabasePath**

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

The desired maximum lock wait time

optimisticLocking (Optional)

Type: [System.Boolean](#)

Use optimistic locking. With optimistic locking, readers are always permitted and multiple updaters are permitted to update the same Databases/Pages but only the first transaction to commit or flush an updated Database/Page will be successful in making the committed change. Other updaters will get an `OptimisticLockingException` if they try to commit/flush updates to the same Pages/Database. See http://en.wikipedia.org/wiki/Optimistic_concurrency_control for further explanation of optimistic locking. If optimistic locking is not enabled then pessimistic locking is used. See: http://en.wikipedia.org/wiki/Concurrency_control

enablePageCache (Optional)

Type: [System.Boolean](#)

If strong reference page caching isn't desired, turn it off by setting to false

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

Determines default value of [Cache](#)

enableDatabaseCache (Optional)

Type: [System.Boolean](#)

Is this session going to enable strong reference [Database](#) caching?

See Also




[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.SessionNoServerShared Properties

The [SessionNoServerShared](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddToIndexInSeperateThread | Allow adding objects to indices to be done in a worker thread instead of in main thread. (Overrides SessionBase.AddToIndexInSeperateThread .) |
|  | Databases | Gets a list of the currently opened databases (Overrides SessionBase.Databases .) |
|  | WriteToDiskInSeperateDatabaseThreads | Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. AddToIndexInSeperateThread must also be enabled for this to work. (Overrides SessionBase.WriteToDiskInSeperateDatabaseThreads .) |

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.AddToIndexInSeperateThread Property

Allow adding objects to indices to be done in a worker thread instead of in main thread.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AddToIndexInSeperateThread { get; set; }
```

VB

```
Public Overrides Property AddToIndexInSeperateThread As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool AddToIndexInSeperateThread {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract AddToIndexInSeperateThread : bool with get, set  
override AddToIndexInSeperateThread : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Databases Property

Gets a list of the currently opened databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IList<Database> Databases { get; }
```

VB

```
Public Overrides ReadOnly Property Databases As IList(Of Database)  
    Get
```

C++

```
public:  
virtual property IList<Database^>^ Databases {  
    IList<Database^>^ get () override;  
}
```

F#

```
abstract Databases : IList<Database> with get  
override Databases : IList<Database> with get
```

Property Value

Type: [IList\(Database\)](#)

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.WriteToDiskInSeperateDatabaseThreads

Property

Allow object serialization and page writes to happen in worker threads, one per database, instead of in main session thread. [AddToIndexInSeperateThread](#) must also be enabled for this to work.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool WriteToDiskInSeperateDatabaseThreads { get; set; }
```

VB

```
Public Overrides Property WriteToDiskInSeperateDatabaseThreads As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool WriteToDiskInSeperateDatabaseThreads {  
    bool get () override;  
    void set (bool value) override;  
}
```

F#

```
abstract WriteToDiskInSeperateDatabaseThreads : bool with get, set  
override WriteToDiskInSeperateDatabaseThreads : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also












[SessionNoServerShared Class](#)















[VelocityDb.Session Namespace](#)













SessionNoServerShared.SessionNoServerShared Methods

The [SessionNoServerShared](#) type exposes the following members.










Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Abort | Transaction control, abort the current transaction (Overrides SessionBase.Abort().) |
|  BeginRead | Transaction control, begin a read only transaction (Overrides SessionBase.BeginRead(Boolean).) |
|  BeginUpdate | Transaction control, begin an update transaction. (Overrides SessionBase.BeginUpdate(Boolean).) |
|  Checkpoint | Same as Commit(Boolean, Boolean) followed by BeginUpdate() (Overrides SessionBase.Checkpoint().) |
|  ClearCache | Clears page cache and closes databases. Avoid using this one for now. (Overrides SessionBase.ClearCache().) |
|  ClearCachedPages | Clear cache of cached pages (Overrides SessionBase.ClearCachedPages().) |
|  ClearPageCache | Clears cached pages from cache including page weak references. (Overrides SessionBase.ClearPageCache().) |
|  Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact().) |
|  Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database).) |
|  CopyAllDatabasesTo | Copies all databases to a selected directory on the local host. (Overrides SessionBase.CopyAllDatabasesTo(String, Boolean).) |
|  CrossTransactionCache | By default databases are only referenced by a WeakReference across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened. (Overrides SessionBase.CrossTransactionCache(Database, Boolean).) |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DeleteLocation | Deletes a DatabaseLocation , location must first not have any Databases in it (Overrides SessionBase.DeleteLocation(DatabaseLocation, Boolean).) |
|  FileOpen | Opens a Database file for read/update. Internal use only. (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean).) |
|  FlushPageOf | By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written. (Overrides SessionBase.FlushPageOf(OptimizedPersistable).) |
|  FlushUpdates | Write all updated data. This may free up some memory. (Overrides SessionBase.FlushUpdates().) |
|  ForceDatabaseCacheValidation | Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction. (Overrides SessionBase.ForceDatabaseCacheValidation().) |
|  GlobalObjWrapperGet | Lookup wrapper object for a non IOptimizedPersistable object (Overrides SessionBase.GlobalObjWrapperGet(Object, IOptimizedPersistable).) |
|  NewDatabase | Create a new Database with a given database number (Overrides SessionBase.NewDatabase(UInt32, UInt32, String, Boolean).) |
|  NewLocation | Creates a new DatabaseLocation or updates existing ones (Overrides SessionBase.NewLocation(DatabaseLocation).) |
|  Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
|  Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |
|  OpenAllDatabases | Open all databases (Overrides SessionBase.OpenAllDatabases(Boolean).) |
|  OpenDatabase | Opens a Database (Overrides SessionBase.OpenDatabase(UInt32, Boolean, Boolean).) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OpenLocationDatabases | Opens all the databases in a given location (Overrides SessionBase.OpenLocationDatabases(DatabaseLocation, Boolean).) |
|  OpenSchema | Get the session active schema (Overrides SessionBase.OpenSchema(Boolean).) |
|  Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)).) |
|  Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)).) |
|  PossiblyFlushUpdatedPages | Call this function if you may have updated many pages to possibly free up memory if too much memory is in use. (Overrides SessionBase.PossiblyFlushUpdatedPages(UInt32).) |
|  RestoreFrom | Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore. (Overrides SessionBase.RestoreFrom(DatabaseLocation, DateTime).) |
|  SubscribeToChanges | Subscribe to committed database changes of instances of a type when an optional property evaluates to true. (Overrides SessionBase.SubscribeToChanges(Type, String).) |
|  Unpersist | Use this when you want to delete objects that are not assignable as IOptimizedPersistable (Overrides SessionBase.Unpersist(Object).) |
|  UpdateDatabase | Request an update lock on a database (Overrides SessionNoServer.UpdateDatabase(Database).) |
|  UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |
|  Verify | Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found. (Overrides SessionBase.Verify().) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  | ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  | ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  | ImportJson(T) | (Defined by JsonImportExport.) |
|  | MicrosoftSync | (Defined by Sync.) |
|  | SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  | SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[SessionNoServerShared Class](#)[VelocityDb.Session Namespace](#)

SessionNoServerShared.Abort Method

Transaction control, abort the current transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Abort ()
```

VB

```
Public Overrides Sub Abort
```

C++

```
public:  
virtual void Abort () override
```

F#

```
abstract Abort : unit -> unit  
override Abort : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.BeginRead Method

Transaction control, begin a read only transaction

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginRead(  
    bool doRecoveryCheck = false  
)
```

VB

```
Public Overrides Function BeginRead (  
    Optional doRecoveryCheck As Boolean = false  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginRead(  
    bool doRecoveryCheck = false  
) override
```

F#

```
abstract BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction  
override BeginRead :  
    ?doRecoveryCheck : bool  
(* Defaults:  
    let _doRecoveryCheck = defaultArg doRecoveryCheck false  
)  
-> SessionBase.Transaction
```

Parameters

doRecoveryCheck (Optional)

Type: [System.Boolean](#)

By default we do not check for a failed update transaction that requires reverting one or more database changes. We don't do it by default because it can be a little costly. Change this parameter to `true` if you prefer ensured consistency over better performance or make change only if you encounter an exception in the transaction.

VelocityDB Class Library

Return Value

Type: [SessionBase.Transaction](#)

A transaction helper object.

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.BeginUpdate Method

Transaction control, begin an update transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction BeginUpdate(  
    bool doRecoveryCheck  
)
```

VB

```
Public Overrides Function BeginUpdate (  
    doRecoveryCheck As Boolean  
) As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ BeginUpdate (  
    bool doRecoveryCheck  
) override
```

F#

```
abstract BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction  
override BeginUpdate :  
    doRecoveryCheck : bool -> SessionBase.Transaction
```

Parameters

doRecoveryCheck

Type: [System.Boolean](#)

Set to false only when moving system databases to a new directory

Return Value

Type: [SessionBase.Transaction](#)

A [SessionBase.Transaction](#) helper object.

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Checkpoint Method

Same as [Commit\(Boolean, Boolean\)](#) followed by [BeginUpdate\(\)](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override SessionBase.Transaction Checkpoint ()
```

VB

```
Public Overrides Function Checkpoint As SessionBase.Transaction
```

C++

```
public:  
virtual SessionBase.Transaction^ Checkpoint () override
```

F#

```
abstract Checkpoint : unit -> SessionBase.Transaction  
override Checkpoint : unit -> SessionBase.Transaction
```

Return Value

Type: [SessionBase.Transaction](#)

[SessionBase.Transaction](#) started

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.ClearCache Method

Clears page cache and closes databases. Avoid using this one for now.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearCache ()
```

VB

```
Public Overrides Sub ClearCache
```

C++

```
public:  
virtual void ClearCache () override
```

F#

```
abstract ClearCache : unit -> unit  
override ClearCache : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.ClearCachedPages Method

Clear cache of cached pages

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearCachedPages ()
```

VB

```
Public Overrides Sub ClearCachedPages
```

C++

```
public:  
virtual void ClearCachedPages () override
```

F#

```
abstract ClearCachedPages : unit -> unit  
override ClearCachedPages : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.ClearPageCache Method

Clears cached pages from cache including page weak references.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ClearPageCache ()
```

VB

```
Public Overrides Sub ClearPageCache
```

C++

```
public:  
virtual void ClearPageCache () override
```

F#

```
abstract ClearPageCache : unit -> unit  
override ClearPageCache : unit -> unit
```



See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Compact Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Compact() | Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each Database file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction. (Overrides SessionBase.Compact().) |
|  | Compact(Database) | Reduce size of database, if possible, by truncating file where unused space begins (Overrides SessionBase.Compact(Database).) |

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Compact Method

Reduce size of databases, if possible, by first attempting to relocate pages to free areas towards the beginning of each [Database](#) file and then by truncating files where unused space begins. Run Compact() outside the scope of any transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact ()
```

VB

```
Public Overrides Sub Compact
```

C++

```
public:  
virtual void Compact () override
```

F#

```
abstract Compact : unit -> unit  
override Compact : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Compact Method (Database)

Reduce size of database, if possible, by truncating file where unused space begins

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Compact (  
    Database db  
)
```

VB

```
Public Overrides Sub Compact (  
    db As Database  
)
```

C++

```
public:  
virtual void Compact (  
    Database^ db  
) override
```

F#

```
abstract Compact :  
    db : Database -> unit  
override Compact :  
    db : Database -> unit
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

to compact

See Also

[SessionNoServerShared Class](#)

[Compact Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.CopyAllDatabasesTo Method

Copies all databases to a selected directory on the local host.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void CopyAllDatabasesTo(  
    string directory,  
    bool systemDatabaseLocationOnly  
)
```

VB

```
Public Overrides Sub CopyAllDatabasesTo (  
    directory As String,  
    systemDatabaseLocationOnly As Boolean  
)
```

C++

```
public:  
virtual void CopyAllDatabasesTo(  
    String^ directory,  
    bool systemDatabaseLocationOnly  
) override
```

F#

```
abstract CopyAllDatabasesTo :  
    directory : string *  
    systemDatabaseLocationOnly : bool -> unit  
override CopyAllDatabasesTo :  
    directory : string *  
    systemDatabaseLocationOnly : bool -> unit
```

Parameters

directory

Type: [System.String](#)

Path to a directory

systemDatabaseLocationOnly

Type: [System.Boolean](#)

[Missing <param name="systemDatabaseLocationOnly"/> documentation for "M:VelocityDb.Session.SessionNoServerShared.CopyAllDatabasesTo(System.String,System.Boolean)"]

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.CrossTransactionCache Method

By default databases are only referenced by a [WeakReference](#) across transaction boundaries. This means that such Database may or may not be available as a cached database depending on garbage collection activity and if such database also has a strong reference. This function lets you add a strong reference to a Database so the cached Database may be used if version wasn't changed by a different thread since prior transaction. The strong reference is removed once the Database is reopened.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void CrossTransactionCache (
    Database db,
    bool enable = true
)
```

VB

```
Public Overrides Sub CrossTransactionCache (
    db As Database,
    Optional enable As Boolean = true
)
```

C++

```
public:
virtual void CrossTransactionCache (
    Database^ db,
    bool enable = true
) override
```

F#

```
abstract CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
override CrossTransactionCache :
    db : Database *
    ?enable : bool
(* Defaults:
    let _enable = defaultArg enable true
*)
-> unit
```

Parameters

db

VelocityDB Class Library

Type: [VelocityDb.Database](#)

The Database to cache

enable (Optional)

Type: [System.Boolean](#)

Add or remove strong reference. If true, add a strong reference

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.DeleteLocation Method

Deletes a [DatabaseLocation](#), location must first not have any Databases in it

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void DeleteLocation(  
    DatabaseLocation location,  
    bool catalogOnly = false  
)
```

VB

```
Public Overrides Sub DeleteLocation (  
    location As DatabaseLocation,  
    Optional catalogOnly As Boolean = false  
)
```

C++

```
public:  
virtual void DeleteLocation(  
    DatabaseLocation^ location,  
    bool catalogOnly = false  
) override
```

F#

```
abstract DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
)  
-> unit  
override DeleteLocation :  
    location : DatabaseLocation *  
    ?catalogOnly : bool  
(* Defaults:  
    let_catalogOnly = defaultArg catalogOnly false  
)  
-> unit
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The DatabaseLocation to delete

catalogOnly (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Of true, only delete catalog entry. Leave the directory as is.

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.FileOpen Method

Opens a [Database](#) file for read/update. Internal use only.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Stream FileOpen(
    Database db,
    FileAccess fileAccess,
    ref string errorMessage,
    FileMode fileMode = FileMode.Open,
    bool exclusiveAccess = false,
    int waitOverride = 0,
    bool signalError = true
)
```

VB

```
Public Overrides Function FileOpen (
    db As Database,
    fileAccess As FileAccess,
    ByRef errorMessage As String,
    Optional fileMode As FileMode = FileMode.Open,
    Optional exclusiveAccess As Boolean = false,
    Optional waitOverride As Integer = 0,
    Optional signalError As Boolean = true
) As Stream
```

C++

```
public:
virtual Stream^ FileOpen(
    Database^ db,
    FileAccess fileAccess,
    String^% errorMessage,
    FileMode fileMode = FileMode::Open,
    bool exclusiveAccess = false,
    int waitOverride = 0,
    bool signalError = true
) override
```

F#

```
abstract FileOpen :
    db : Database *
    fileAccess : FileAccess *
    errorMessage : string byref *
    ?fileMode : FileMode *
    ?exclusiveAccess : bool *
    ?waitOverride : int *
    ?signalError : bool
```

```
(* Defaults:
    let_fileMode = defaultArg fileMode FileMode.Open
    let_exclusiveAccess = defaultArg exclusiveAccess false
    let_waitOverride = defaultArg waitOverride 0
    let_signalError = defaultArg signalError true
*)
-> Stream
override FileOpen :
    db : Database *
    fileAccess : FileAccess *
    errorMessage : string byref *
    ?fileMode : FileMode *
    ?exclusiveAccess : bool *
    ?waitOverride : int *
    ?signalError : bool
(* Defaults:
    let_fileMode = defaultArg fileMode FileMode.Open
    let_exclusiveAccess = defaultArg exclusiveAccess false
    let_waitOverride = defaultArg waitOverride 0
    let_signalError = defaultArg signalError true
*)
-> Stream
```

Parameters

db

Type: [VelocityDb.Database](#)

[Database](#)

for which we want to open [Stream](#)

fileAccess

Type: [System.IO.FileAccess](#)

Opening for read or update?

errorMessage

Type: [System.String](#)

Used for passing error messages back to caller

fileMode (Optional)

Type: [System.IO.FileMode](#)

[FileMode](#)

used for opening file

exclusiveAccess (Optional)

Type: [System.Boolean](#)

Do we want exclusive access to file?

waitOverride (Optional)

Type: [System.Int32](#)

How long are we willing to wait for a file lock to clear?

VelocityDB Class Library

signalError (Optional)

Type: [System.Boolean](#)

Throw exception if we can't open file within wait time?

Return Value

Type: [Stream](#)

A [Stream](#) used for accessing [Database](#) data

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.FlushPageOf Method

By calling this you force a persisted (has an Id) object to be written to disk (if updated) and indices (if any) to be updated. Other objects on the same page page will also be written.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushPageOf(  
    OptimizedPersistable pObj  
)
```

VB

```
Public Overrides Sub FlushPageOf (  
    pObj As OptimizedPersistable  
)
```

C++

```
public:  
virtual void FlushPageOf(  
    OptimizedPersistable^ pObj  
) override
```

F#

```
abstract FlushPageOf :  
    pObj : OptimizedPersistable -> unit  
override FlushPageOf :  
    pObj : OptimizedPersistable -> unit
```

Parameters

pObj

Type: [VelocityDb.OptimizedPersistable](#)

An object indicating what [Page](#) to flush

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.FlushUpdates Method

Write all updated data. This may free up some memory.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void FlushUpdates ()
```

VB

```
Public Overrides Sub FlushUpdates
```

C++

```
public:  
virtual void FlushUpdates () override
```

F#

```
abstract FlushUpdates : unit -> unit  
override FlushUpdates : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.ForceDatabaseCacheValidation Method

Cached data is set to be validated whenever a new transaction is started. This function is provided as a way to force cache validation within a transaction without requiring a commit followed by a new transaction.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ForceDatabaseCacheValidation()
```

VB

```
Public Overrides Sub ForceDatabaseCacheValidation
```

C++

```
public:  
virtual void ForceDatabaseCacheValidation() override
```

F#

```
abstract ForceDatabaseCacheValidation : unit -> unit  
override ForceDatabaseCacheValidation : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.GlobalObjWrapperGet Method

Lookup wrapper object for a non IOptimizedPersistable object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool GlobalObjWrapperGet (
    Object lookupObj,
    out IOptimizedPersistable pObj
)
```

VB

```
Public Overrides Function GlobalObjWrapperGet (
    lookupObj As Object,
    <OutAttribute> ByRef pObj As IOptimizedPersistable
) As Boolean
```

C++

```
public:
virtual bool GlobalObjWrapperGet (
    Object^ lookupObj,
    [OutAttribute] IOptimizedPersistable^% pObj
) override
```

F#

```
abstract GlobalObjWrapperGet :
    lookupObj : Object *
    pObj : IOptimizedPersistable byref -> bool
override GlobalObjWrapperGet :
    lookupObj : Object *
    pObj : IOptimizedPersistable byref -> bool
```

Parameters

lookupObj

Type: [System.Object](#)

object to look for

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

wrapper object

Return Value

Type: [Boolean](#)

Wrapper object

VelocityDB Class Library

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.NewDatabase Method

Create a new Database with a given database number

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database NewDatabase(
    uint dbNum,
    uint megaBytesPresize = 0,
    string name = null,
    bool signalError = true
)
```

VB

```
Public Overrides Function NewDatabase (
    dbNum As UInteger,
    Optional megaBytesPresize As UInteger = 0,
    Optional name As String = Nothing,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ NewDatabase(
    unsigned int dbNum,
    unsigned int megaBytesPresize = 0,
    String^ name = nullptr,
    bool signalError = true
) override
```

F#

```
abstract NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
    let_megaBytesPresize = defaultArg megaBytesPresize 0
    let_name = defaultArg name null
    let_signalError = defaultArg signalError true
*)
-> Database
override NewDatabase :
    dbNum : uint32 *
    ?megaBytesPresize : uint32 *
    ?name : string *
    ?signalError : bool
(* Defaults:
```

VelocityDB Class Library

```
        let_megaBytesPresize = defaultArg megaBytesPresize 0
        let_name = defaultArg name null
        let_signalError = defaultArg signalError true
*)
-> Database
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the database to create

megaBytesPresize (Optional)

Type: [System.UInt32](#)

If you know that the Database will be large, presizing it may avoid file fragmentation. Default value is 0

name (Optional)

Type: [System.String](#)

Optionally name the new [Database](#)

signalError (Optional)

Type: [System.Boolean](#)

Optionally signal an error if creation of new [Database](#) fails

Return Value

Type: [Database](#)

The newly created [Database](#) or throws an exception if the [Database](#) already exist

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.NewLocation Method

Creates a new DatabaseLocation or updates existing ones

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override DatabaseLocation NewLocation(  
    DatabaseLocation location  
)
```

VB

```
Public Overrides Function NewLocation (  
    location As DatabaseLocation  
) As DatabaseLocation
```

C++

```
public:  
virtual DatabaseLocation^ NewLocation(  
    DatabaseLocation^ location  
) override
```

F#

```
abstract NewLocation :  
    location : DatabaseLocation -> DatabaseLocation  
override NewLocation :  
    location : DatabaseLocation -> DatabaseLocation
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The input location

Return Value

Type: [DatabaseLocation](#)

The new DatabaseLocation or an existing one





See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Open Method

Overload List

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Open(UInt64, Boolean, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt64, Boolean, Boolean, Int32).) |
|  Open(UInt32, UInt16, UInt16, Boolean, Int32) | Opens a persistent object (Overrides SessionBase.Open(UInt32, UInt16, UInt16, Boolean, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32) | For internal usage only, used in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32).) |
|  Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean) | For internal use only in generated code (Overrides SessionBase.Open(Database, UInt64, Boolean, List(IOptimizedPersistable), Boolean, Int32, Int32, Boolean).) |

See Also

[SessionNoServerShared Class](#)[VelocityDb.Session Namespace](#)

SessionNoServerShared.Open Method (UInt64, Boolean, Boolean, Int32)

Opens a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Open(
    ulong oid,
    bool update,
    bool inFlush,
    int graphDepthToLoad = 2147483647
)
```

VB

```
Public Overrides Function Open (
    oid As ULong,
    update As Boolean,
    inFlush As Boolean,
    Optional graphDepthToLoad As Integer = 2147483647
) As Object
```

C++

```
public:
virtual Object^ Open(
    unsigned long long oid,
    bool update,
    bool inFlush,
    int graphDepthToLoad = 2147483647
) override
```

F#

```
abstract Open :
    oid : uint64 *
    update : bool *
    inFlush : bool *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> Object
override Open :
    oid : uint64 *
    update : bool *
    inFlush : bool *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
```

```
* )  
-> Object
```

Parameters

oid

Type: [System.UInt64](#)

The Id of the object to open

update

Type: [System.Boolean](#)

Open the object for update?

inFlush

Type: [System.Boolean](#)

if true, disallow page flushing while opening the object

graphDepthToLoad (Optional)

Type: [System.Int32](#)

Set this if you want to limit the depth of the graph loaded by this open.

Return Value

Type: [Object](#)

A persistent object

See Also

[SessionNoServerShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Open Method (UInt32, UInt16, UInt16, Boolean, Int32)

Opens a persistent object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(  
    uint dbNum,  
    ushort pageNum,  
    ushort slotNum,  
    bool update,  
    int graphDepthToLoad = 2147483647  
)
```

VB

```
Public Overrides Function Open (  
    dbNum As UInteger,  
    pageNum As UShort,  
    slotNum As UShort,  
    update As Boolean,  
    Optional graphDepthToLoad As Integer = 2147483647  
) As IOptimizedPersistable
```

C++

```
public:  
virtual IOptimizedPersistable^ Open(  
    unsigned int dbNum,  
    unsigned short pageNum,  
    unsigned short slotNum,  
    bool update,  
    int graphDepthToLoad = 2147483647  
) override
```

F#

```
abstract Open :  
    dbNum : uint32 *  
    pageNum : uint16 *  
    slotNum : uint16 *  
    update : bool *  
    ?graphDepthToLoad : int  
(* Defaults:  
    let graphDepthToLoad = defaultArg graphDepthToLoad 2147483647  
)  
-> IOptimizedPersistable  
override Open :  
    dbNum : uint32 *
```

VelocityDB Class Library

```
    pageNum : uint16 *
    slotNum : uint16 *
    update : bool *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
```

Parameters

dbNum

Type: [System.UInt32](#)

Database number of the object to open

pageNum

Type: [System.UInt16](#)

Page number of the object to open

slotNum

Type: [System.UInt16](#)

Slot number of the object to open

update

Type: [System.Boolean](#)

Open the object for update?

graphDepthToLoad (Optional)

Type: [System.Int32](#)

Limit depth of graph to open

Return Value

Type: [IOptimizedPersistable](#)

The opened object or null if it does not exist

See Also

[SessionNoServerShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Open Method (Database, UInt64, Boolean, List<IOptimizedPersistable>, Boolean, Int32, Int32)

For internal usage only, used in generated code

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(
    Database db,
    ulong oid,
    bool update,
    List<IOptimizedPersistable> toLoadMembers,
    bool inFlush,
    int graphDepth = 0,
    int graphDepthToLoad = 2147483647
)
```

VB

```
Public Overrides Function Open (
    db As Database,
    oid As ULong,
    update As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    inFlush As Boolean,
    Optional graphDepth As Integer = 0,
    Optional graphDepthToLoad As Integer = 2147483647
) As IOptimizedPersistable
```

C++

```
public:
virtual IOptimizedPersistable^ Open(
    Database^ db,
    unsigned long long oid,
    bool update,
    List<IOptimizedPersistable^>^ toLoadMembers,
    bool inFlush,
    int graphDepth = 0,
    int graphDepthToLoad = 2147483647
) override
```

F#

```
abstract Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
```

```
        ?graphDepth : int *
        ?graphDepthToLoad : int
(* Defaults:
    let_graphDepth = defaultArg graphDepth 0
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
override Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
    ?graphDepth : int *
    ?graphDepthToLoad : int
(* Defaults:
    let_graphDepth = defaultArg graphDepth 0
    let_graphDepthToLoad = defaultArg graphDepthToLoad 2147483647
*)
-> IOptimizedPersistable
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

oid

Type: [System.UInt64](#)

[Missing <param name="oid"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

update

Type: [System.Boolean](#)

[Missing <param name="update"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

inFlush

Type: [System.Boolean](#)

[Missing <param name="inFlush"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

graphDepth (Optional)

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

graphDepthToLoad (Optional)

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

Return Value

Type: [IOptimizedPersistable](#)

[Missing <returns> documentation for

"M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32)"]

See Also

[SessionNoServerShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Open Method (Database, UInt64, Boolean, List<IOptimizedPersistable>, Boolean, Int32, Int32, Boolean)

For internal use only in generated code

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IOptimizedPersistable Open(  
    Database db,  
    ulong oid,  
    bool update,  
    List<IOptimizedPersistable> toLoadMembers,  
    bool inFlush,  
    int graphDepth,  
    int graphDepthToLoad,  
    bool signalNotExistError = false  
)
```

VB

```
Public Overrides Function Open (  
    db As Database,  
    oid As ULong,  
    update As Boolean,  
    toLoadMembers As List(Of IOptimizedPersistable),  
    inFlush As Boolean,  
    graphDepth As Integer,  
    graphDepthToLoad As Integer,  
    Optional signalNotExistError As Boolean = false  
) As IOptimizedPersistable
```

C++

```
public:  
virtual IOptimizedPersistable^ Open(  
    Database^ db,  
    unsigned long long oid,  
    bool update,  
    List<IOptimizedPersistable^>^ toLoadMembers,  
    bool inFlush,  
    int graphDepth,  
    int graphDepthToLoad,  
    bool signalNotExistError = false  
) override
```

F#

```
abstract Open :  
    db : Database *  
    oid : uint64 *
```

```
        update : bool *
        toLoadMembers : List<IOptimizedPersistable> *
        inFlush : bool *
        graphDepth : int *
        graphDepthToLoad : int *
        ?signalNotExistError : bool
(* Defaults:
    let_signalNotExistError = defaultArg signalNotExistError false
*)
-> IOptimizedPersistable
override Open :
    db : Database *
    oid : uint64 *
    update : bool *
    toLoadMembers : List<IOptimizedPersistable> *
    inFlush : bool *
    graphDepth : int *
    graphDepthToLoad : int *
    ?signalNotExistError : bool
(* Defaults:
    let_signalNotExistError = defaultArg signalNotExistError false
*)
-> IOptimizedPersistable
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDb.Session.SessionNoServerShared.Open(VelocityDb.Database,System.UInt64,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Boolean,System.Int32,System.Int32,System.Boolean)"]

oid

Type: [System.UInt64](#)

Id of the object to open, must match with Database id

update

Type: [System.Boolean](#)

Update object when opening it

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

Internal list of objects to be loaded as part of graph load

inFlush

Type: [System.Boolean](#)

Permit page flushes as side affect of opening object?

graphDepth

Type: [System.Int32](#)

Current graph depth of graph load

VelocityDB Class Library

graphDepthToLoad

Type: [System.Int32](#)

Max graph depth to load

signalNotExistError (Optional)

Type: [System.Boolean](#)

Signal an exception if object/page/database does not exist.

Return Value

Type: [IOptimizedPersistable](#)

a persistent object

See Also

[SessionNoServerShared Class](#)

[Open Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.OpenAllDatabases Method

Open all databases

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
) override
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

Open for update?

Return Value

Type: [List\(Database\)](#)

List of [Database](#)

See Also

[SessionNoServerShared Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionNoServerShared.OpenDatabase Method

Opens a Database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Database OpenDatabase(
    uint dbNum,
    bool update = false,
    bool signalError = true
)
```

VB

```
Public Overrides Function OpenDatabase (
    dbNum As UInteger,
    Optional update As Boolean = false,
    Optional signalError As Boolean = true
) As Database
```

C++

```
public:
virtual Database^ OpenDatabase(
    unsigned int dbNum,
    bool update = false,
    bool signalError = true
) override
```

F#

```
abstract OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
override OpenDatabase :
    dbNum : uint32 *
    ?update : bool *
    ?signalError : bool
(* Defaults:
    let_update = defaultArg update false
    let_signalError = defaultArg signalError true
*)
-> Database
```

VelocityDB Class Library

Parameters

dbNum

Type: [System.UInt32](#)

The Database number of the database to open

update (Optional)

Type: [System.Boolean](#)

Open the Database for update?

signalError (Optional)

Type: [System.Boolean](#)

Signal errors if errors found when opening the Database?

Return Value

Type: [Database](#)

The opened Database

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.OpenLocationDatabases Method

Opens all the databases in a given location

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override List<Database> OpenLocationDatabases (  
    DatabaseLocation location,  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenLocationDatabases (  
    location As DatabaseLocation,  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenLocationDatabases (  
    DatabaseLocation^ location,  
    bool update = false  
) override
```

F#

```
abstract OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenLocationDatabases :  
    location : DatabaseLocation *  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

The location for which to open databases

update (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

Shall each database be opened for update?

Return Value

Type: [List\(Database\)](#)

A list of opened databases

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.OpenSchema Method

Get the session active schema

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override Schema OpenSchema (  
    bool update  
)
```

```
VB  
Public Overrides Function OpenSchema (  
    update As Boolean  
) As Schema
```

```
C++  
public:  
virtual Schema^ OpenSchema (  
    bool update  
) override
```

```
F#  
abstract OpenSchema :  
    update : bool -> Schema  
override OpenSchema :  
    update : bool -> Schema
```

Parameters

update

Type: [System.Boolean](#)

Do update the schema?

Return Value

Type: [Schema](#)

The active schema



See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Persist Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist(IOptimizedPersistable, Nullable(UInt16)) | This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides PlacementDatabaseNumber and returns something other than DefaultPlacementDatabaseNumber . (Overrides SessionBase.Persist(IOptimizedPersistable, Nullable(UInt16)) .) |
|  | Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) | Persists an object (Overrides SessionBase.Persist(Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable)) .) |

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Persist Method (IOptimizedPersistable, Nullable(UInt16))

This is the recommended way of persisting objects, it is simple and efficient. Each type is stored in its own database unless object class overrides [PlacementDatabaseNumber](#) and returns something other than [DefaultPlacementDatabaseNumber](#).

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    IOptimizedPersistable ipObj,
    ushort? objectsPerPage = null
)
```

VB

```
Public Overrides Function Persist (
    ipObj As IOptimizedPersistable,
    Optional objectsPerPage As UShort? = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    IOptimizedPersistable^ ipObj,
    Nullable<unsigned short> objectsPerPage = nullptr
) override
```

F#

```
abstract Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let _objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
override Persist :
    ipObj : IOptimizedPersistable *
    ?objectsPerPage : Nullable<uint16>
(* Defaults:
    let _objectsPerPage = defaultArg objectsPerPage null
*)
-> uint64
```

Parameters

ipObj

VelocityDB Class Library

Type: [VelocityDb.IOptimizedPersistable](#)

The object to make persistent

objectsPerPage (Optional)

Type: [System.Nullable\(UInt16\)](#)

Override of objects per page, only respected when persisting first object of some type

Return Value

Type: [UInt64](#)

The Id of the persisted object

See Also

[SessionNoServerShared Class](#)

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Persist Method (Placement, IOptimizedPersistable, Schema, UInt16, Boolean, Queue(IOptimizedPersistable))

Persists an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    IOptimizedPersistable pObj,
    Schema schema,
    ushort slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    pObj As IOptimizedPersistable,
    schema As Schema,
    Optional slotLimitPerPage As UShort = 0,
    Optional inFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    IOptimizedPersistable^ pObj,
    Schema^ schema,
    unsigned short slotLimitPerPage = 0,
    bool inFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
```

```
(* Defaults:
    let_slotLimitPerPage = defaultArg slotLimitPerPage 0
    let_inFlush = defaultArg inFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
    place : Placement *
    pObj : IOptimizedPersistable *
    schema : Schema *
    ?slotLimitPerPage : uint16 *
    ?inFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_slotLimitPerPage = defaultArg slotLimitPerPage 0
    let_inFlush = defaultArg inFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

Placement object determining where to persist the object

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

The object to persist

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active session schema

slotLimitPerPage (Optional)

Type: [System.UInt16](#)

Override of how many slots per page to permit

inFlush (Optional)

Type: [System.Boolean](#)

Set to true to disallow page flushes as a side affect

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A list of objects waiting to be persisted

Return Value

Type: [UInt64](#)

Id of persisted object

See Also

[SessionNoServerShared Class](#)

VelocityDB Class Library

[Persist Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.PossiblyFlushUpdatedPages Method

Call this function if you may have updated many pages to possibly free up memory if too much memory is in use.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void PossiblyFlushUpdatedPages (
    uint numberOfUpdatedPagesPermittedWithoutFlushCheck = 100
)
```

VB

```
Public Overrides Sub PossiblyFlushUpdatedPages (
    Optional numberOfUpdatedPagesPermittedWithoutFlushCheck As UInteger =
100
)
```

C++

```
public:
virtual void PossiblyFlushUpdatedPages (
    unsigned int numberOfUpdatedPagesPermittedWithoutFlushCheck = 100
) override
```

F#

```
abstract PossiblyFlushUpdatedPages :
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32
(* Defaults:
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg
numberOfUpdatedPagesPermittedWithoutFlushCheck 100
*)
-> unit
override PossiblyFlushUpdatedPages :
    ?numberOfUpdatedPagesPermittedWithoutFlushCheck : uint32
(* Defaults:
    let numberOfUpdatedPagesPermittedWithoutFlushCheck = defaultArg
numberOfUpdatedPagesPermittedWithoutFlushCheck 100
*)
-> unit
```

Parameters

numberOfUpdatedPagesPermittedWithoutFlushCheck (Optional)

Type: [System.UInt32](#)

Only do check if the number of updated pages is greater than this number

See Also

[SessionNoServerShared Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

SessionNoServerShared.RestoreFrom Method

Restores Databases and pages from a backup DatabaseLocation. Existing data will be merged with the restored data unless existing Databases to restore are deleted before the restore.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void RestoreFrom(  
    DatabaseLocation backupLocation,  
    DateTime upToTime  
)
```

VB

```
Public Overrides Sub RestoreFrom (  
    backupLocation As DatabaseLocation,  
    upToTime As DateTime  
)
```

C++

```
public:  
virtual void RestoreFrom(  
    DatabaseLocation^ backupLocation,  
    DateTime upToTime  
) override
```

F#

```
abstract RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit  
override RestoreFrom :  
    backupLocation : DatabaseLocation *  
    upToTime : DateTime -> unit
```

Parameters

backupLocation

Type: [VelocityDb.DatabaseLocation](#)

The location to restore from

upToTime

Type: [System.DateTime](#)

Restore location up to a given DateTime. Data backed up after this time will not be restored.

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.SubscribeToChanges Method

Subscribe to committed database changes of instances of a type when an optional property evaluates to true.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SubscribeToChanges (
    Type aType,
    string notifyIfTrueProperty = null
)
```

VB

```
Public Overrides Sub SubscribeToChanges (
    aType As Type,
    Optional notifyIfTrueProperty As String = Nothing
)
```

C++

```
public:
virtual void SubscribeToChanges (
    Type^ aType,
    String^ notifyIfTrueProperty = nullptr
) override
```

F#

```
abstract SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
override SubscribeToChanges :
    aType : Type *
    ?notifyIfTrueProperty : string
(* Defaults:
    let_notifyIfTrueProperty = defaultArg notifyIfTrueProperty null
*)
-> unit
```

Parameters

aType

Type: [System.Type](#)

The type you are interested in being notified about when persistent instances changes within a database.

notifyIfTrueProperty (Optional)

Type: [System.String](#)

The name of a property part of the type specified as aType, this should be a Boolean property. When property returns true, a change notification is send otherwise no notification is send. Leave as null to get a notification any time any instance of the selected type changes

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Unpersist Method

Use this when you want to delete objects that are not assignable as [IOptimizedPersistable](#)

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override bool Unpersist(  
    Object obj  
)
```

```
VB  
Public Overrides Function Unpersist (  
    obj As Object  
) As Boolean
```

```
C++  
public:  
virtual bool Unpersist(  
    Object^ obj  
) override
```

```
F#  
abstract Unpersist :  
    obj : Object -> bool  
override Unpersist :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to be deleted from persistent storage

Return Value

Type: [Boolean](#)

true if object found and deleted; otherwise false

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.UpdateDatabase Method

Request an update lock on a database

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override bool UpdateDatabase(  
    Database db  
)
```

```
VB  
Public Overrides Function UpdateDatabase (  
    db As Database  
) As Boolean
```

```
C++  
public:  
virtual bool UpdateDatabase(  
    Database^ db  
) override
```

```
F#  
abstract UpdateDatabase :  
    db : Database -> bool  
override UpdateDatabase :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

The database to update

Return Value

Type: [Boolean](#)

true if Database was updated, otherwise throws an exception



See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.UpdateObject Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  | UpdateObject(Object) | Tag an object as updated so that it will be updated persistently (Overrides SessionBase.UpdateObject(Object).) |
|  | UpdateObject(IOptimizedPersistable, Boolean, Boolean) | Updates an object (Overrides SessionBase.UpdateObject(IOptimizedPersistable, Boolean, Boolean).) |

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.UpdateObject Method (Object)

Tag an object as updated so that it will be updated persistently

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public override bool UpdateObject(  
    Object obj  
)
```

```
VB  
Public Overrides Function UpdateObject (  
    obj As Object  
) As Boolean
```

```
C++  
public:  
virtual bool UpdateObject(  
    Object^ obj  
) override
```

```
F#  
abstract UpdateObject :  
    obj : Object -> bool  
override UpdateObject :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to update.

Return Value

Type: [Boolean](#)

true if object updated; otherwise false

See Also

[SessionNoServerShared Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.UpdateObject Method (IOptimizedPersistable, Boolean, Boolean)

Updates an object

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool UpdateObject(  
    IOptimizedPersistable obj,  
    bool inFlush,  
    bool deleteObjFromIndexes = true  
)
```

VB

```
Public Overrides Function UpdateObject (  
    obj As IOptimizedPersistable,  
    inFlush As Boolean,  
    Optional deleteObjFromIndexes As Boolean = true  
) As Boolean
```

C++

```
public:  
virtual bool UpdateObject(  
    IOptimizedPersistable^ obj,  
    bool inFlush,  
    bool deleteObjFromIndexes = true  
) override
```

F#

```
abstract UpdateObject :  
    obj : IOptimizedPersistable *  
    inFlush : bool *  
    ?deleteObjFromIndexes : bool  
(* Defaults:  
    let_deleteObjFromIndexes = defaultArg deleteObjFromIndexes true  
)  
-> bool  
override UpdateObject :  
    obj : IOptimizedPersistable *  
    inFlush : bool *  
    ?deleteObjFromIndexes : bool  
(* Defaults:  
    let_deleteObjFromIndexes = defaultArg deleteObjFromIndexes true  
)  
-> bool
```

VelocityDB Class Library

Parameters

obj

Type: [VelocityDb.IOptimizedPersistable](#)

Object to be updated

inFlush

Type: [System.Boolean](#)

Are we doing this as part of a page flush? Set to true to avoid a recursive page flush

deleteObjFromIndexes (Optional)

Type: [System.Boolean](#)

Set to false if you know that this object is not part of any index. Safe to leave set to true in any case

Return Value

Type: [Boolean](#)

True if object is updated; otherwise false

See Also

[SessionNoServerShared Class](#)

[UpdateObject Overload](#)

[VelocityDb.Session Namespace](#)

SessionNoServerShared.Verify Method

Verifies that databases are valid by reading and following references. An exception is thrown if an issue is found.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Verify()
```

VB

```
Public Overrides Sub Verify
```

C++

```
public:  
virtual void Verify() override
```

F#

```
abstract Verify : unit -> unit  
override Verify : unit -> unit
```

See Also

[SessionNoServerShared Class](#)

[VelocityDb.Session Namespace](#)

SessionPool Class

Provides a pool of sessions for reuse instead of creating new session instances every time a session is needed. The pooled sessions includes cached databases, pages and objects. Keep the pool size to a minimum too avoid excessive memory usage.

Inheritance Hierarchy

[System.Object](#)

VelocityDb.Session.SessionPool

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class SessionPool : IDisposable,
    IComparable<SessionPool>
```

VB

```
Public Class SessionPool
    Implements IDisposable, IComparable(Of SessionPool)
```

C++


```
public ref class SessionPool : IDisposable,
    IComparable<SessionPool^>
```

F#


```
type SessionPool =
    class
        interface IDisposable
        interface IComparable<SessionPool>
    end
```

The **SessionPool** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------------------------------------------------------------------------------------------|
|  | SessionPool | Creates a session pool, store this pool somewhere where your application threads can get to it. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|------------------------------------|
|  | Dispose | Disposes all sessions in this pool |

VelocityDB Class Library

| | |
|---------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
|  FreeSession | Always call this function so that your session is unlocked and becomes available to others |
|  GetSession | Get a session locked for use by caller. |

See Also

[VelocityDb.Session Namespace](#)

SessionPool Constructor

Creates a session pool, store this pool somewhere where your application threads can get to it.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionPool(  
    int poolSize,  
    Func<SessionBase> createSessionFunc  
)
```

VB

```
Public Sub New (  
    poolSize As Integer,  
    createSessionFunc As Func(Of SessionBase)  
)
```

C++

```
public:  
SessionPool(  
    int poolSize,  
    Func<SessionBase^>^ createSessionFunc  
)
```

F#

```
new :  
    poolSize : int *  
    createSessionFunc : Func<SessionBase> -> SessionPool
```

Parameters

poolSize

Type: [System.Int32](#)

The maximum number of session maintained in the pool

createSessionFunc

Type: [System.Func\(SessionBase\)](#)

A function provided by you that creates a session of your choice with your choice of parameters. Could be something as simple as: `() => new SessionNoServer("MyDbDir")`

See Also




[SessionPool Class](#)

[VelocityDb.Session Namespace](#)

SessionPool.SessionPool Methods

The [SessionPool](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------|
|  | Dispose | Disposes all sessions in this pool |
|  | FreeSession | Always call this function so that your session is unlocked and becomes available to others |
|  | GetSession | Get a session locked for use by caller. |

See Also

[SessionPool Class](#)

[VelocityDb.Session Namespace](#)

SessionPool.Dispose Method

Disposes all sessions in this pool

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[SessionPool Class](#)

[VelocityDb.Session Namespace](#)

SessionPool.FreeSession Method

Always call this function so that your session is unlocked and becomes available to others

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FreeSession(  
    int identifier,  
    SessionBase session  
)
```

VB

```
Public Sub FreeSession (  
    identifier As Integer,  
    session As SessionBase  
)
```

C++

```
public:  
void FreeSession(  
    int identifier,  
    SessionBase^ session  
)
```

F#

```
member FreeSession :  
    identifier : int *  
    session : SessionBase -> unit
```

Parameters

identifier

Type: [System.Int32](#)

The identifier provided by [GetSession\(Int32\)](#)

session

Type: [VelocityDb.Session.SessionBase](#)

The session provided by [GetSession\(Int32\)](#)

See Also

[SessionPool Class](#)

[VelocityDb.Session Namespace](#)

SessionPool.GetSession Method

Get a session locked for use by caller.

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public SessionBase GetSession(  
    out int identifier  
)
```

VB

```
Public Function GetSession (  
    <OutAttribute> ByRef identifier As Integer  
) As SessionBase
```

C++

```
public:  
SessionBase^ GetSession(  
    [OutAttribute] int% identifier  
)
```

F#

```
member GetSession :  
    identifier : int byref -> SessionBase
```

Parameters

identifier

Type: [System.Int32](#)

The id of the returned session. Make sure you always call FreeSession with this id provided or else session is locked forever. One way is to call FreeSession in a finally block.

Return Value

Type: [SessionBase](#)

A reserved/locked session

See Also

[SessionPool Class](#)

[VelocityDb.Session Namespace](#)

TypeExtensions Class

Some extensions to Type

[Inheritance Hierarchy](#)

[System.Object](#)

VelocityDb.Session.TypeExtensions

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class TypeExtensions
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class TypeExtensions
```

C++







```
[ExtensionAttribute]  
public ref class TypeExtensions abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type TypeExtensions = class end
```

The **TypeExtensions** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------------|
|   | ExpandsToNonPrimitiveTypes | Used by Database Manager |
|   | GetTypeCode | |
|   | ToGenericTypeString | Gets a string representation of a Type corresponding to how it looks in C# code |







See Also

[VelocityDb.Session Namespace](#)

TypeExtensions.TypeExtensions Methods

The [TypeExtensions](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------------|
|   | ExpandsToNonPrimitiveTypes | Used by Database Manager |
|   | GetTypeCode | |
|   | ToGenericTypeString | Gets a string representation of a Type corresponding to how it looks in C# code |

See Also

[TypeExtensions Class](#)

[VelocityDb.Session Namespace](#)

TypeExtensions.ExpandsToNonPrimitiveTypes Method

Used by Database Manager

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool ExpandsToNonPrimitiveTypes (  
    this Type t  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function ExpandsToNonPrimitiveTypes (  
    t As Type  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool ExpandsToNonPrimitiveTypes (  
    Type^ t  
)
```

F#

```
[<ExtensionAttribute>]  
static member ExpandsToNonPrimitiveTypes :  
    t : Type -> bool
```

Parameters

t

Type: [System.Type](#)

Type examined

Return Value

Type: [Boolean](#)

true if type does; otherwise false

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Type](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[TypeExtensions Class](#)

VelocityDB Class Library

[VelocityDb.Session Namespace](#)

TypeExtensions.GetTypeCode Method

[Missing <summary> documentation for "M:VelocityDb.Session.TypeExtensions.GetTypeCode(System.Type)"]

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TypeCode GetTypeCode (  
    this Type t  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function GetTypeCode (  
    t As Type  
) As TypeCode
```

C++

```
public:  
[ExtensionAttribute]  
static TypeCode GetTypeCode (  
    Type^ t  
)
```

F#

```
[<ExtensionAttribute>]  
static member GetTypeCode :  
    t : Type -> TypeCode
```

Parameters

t

Type: [System.Type](#)

[Missing <param name="t"/> documentation for "M:VelocityDb.Session.TypeExtensions.GetTypeCode(System.Type)"]

Return Value

Type: [TypeCode](#)

[Missing <returns> documentation for "M:VelocityDb.Session.TypeExtensions.GetTypeCode(System.Type)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Type](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[TypeExtensions Class](#)

[VelocityDb.Session Namespace](#)

TypeExtensions.ToGenericTypeString Method

Gets a string representation of a Type corresponding to how it looks in C# code

Namespace: [VelocityDb.Session](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string ToGenericTypeString(  
    this Type t,  
    params Type[] arg  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function ToGenericTypeString (  
    t As Type,  
    ParamArray arg As Type ()  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ ToGenericTypeString(  
    Type^ t,  
    ... array<Type^>^ arg  
)
```

F#

```
[<ExtensionAttribute>]  
static member ToGenericTypeString :  
    t : Type *  
    arg : Type[] -> string
```

Parameters

t

Type: [System.Type](#)

The type to get the string for

arg

Type: [System.Type\[\]](#)

Internally used parameter (optional)

Return Value

Type: [String](#)

A string representation of a type

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Type](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also





[TypeExtensions Class](#)

[VelocityDb.Session Namespace](#)

VelocityDb.Sync Namespace

The `VelocityDb.Sync` namespace contains classes for enabling data sync between different sets of databases

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------|
|  | Change | Used for tracking changes for use with syncing database sessions |
|  | Changes | Keeps track of changes within a set of databases (a federation). Preliminary version. |
|  | ReplicaSync | Contains info of latest transaction synced with databases at a specified host and path |
|  | TransactionChanges | A log of all changes made within a certain transaction. |

Change Class

Used for tracking changes for use with syncing database sessions

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Sync.Change

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class Change : OptimizedPersistable, IComparable<Change>,
    IComparable
```

VB

```
<SerializableAttribute>
Public Class Change
    Inherits OptimizedPersistable
    Implements IComparable(Of Change), IComparable
```

C++


```
[SerializableAttribute]
public ref class Change : public OptimizedPersistable,
    IComparable<Change^>, IComparable
```





F#

```
[<SerializableAttribute>]
type Change =
    class
        inherit OptimizedPersistable
        interface IComparable<Change>
        interface IComparable
    end
```



The **Change** type exposes the following members.

Properties



| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |

| | |
|--------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|
|  DatabaseId | Database number of changed Database |
|  Deleted | Is this a change involving a database/page delete |
|  PageId | Page number of changed Page |
|  Version | Version of page being updated |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareTo(Object) | Compares objects using the object Id (Overrides OptimizedPersistable.CompareTo(Object) .) |
|  | CompareTo(Change) | Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |






See Also

[VelocityDb.Sync Namespace](#)

Change.Change Properties

The [Change](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | DatabaseId | Database number of changed Database |
|  | Deleted | Is this a change involving a database/page delete |
|  | PageId | Page number of changed Page |
|  | Version | Version of page being updated |

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.AllowOtherTypesOnSamePage Property

We only want this type of object on any page containing this type so don't use pages with other type of objects when placing an object of this type

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.DatabaseId Property

Database number of changed [Database](#)

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public uint DatabaseId { get; }
```

VB

```
Public ReadOnly Property DatabaseId As UInteger  
    Get
```

C++

```
public:  
property unsigned int DatabaseId {  
    unsigned int get ();  
}
```

F#

```
member DatabaseId : uint32 with get
```

Property Value

Type: [UInt32](#)

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.Deleted Property

Is this a change involving a database/page delete

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Deleted { get; set; }
```

VB

```
Public Property Deleted As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool Deleted {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member Deleted : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.PageId Property

Page number of changed [Page](#)

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort PageId { get; }
```

VB

```
Public ReadOnly Property PageId As UShort  
    Get
```

C++

```
public:  
property unsigned short PageId {  
    unsigned short get ();  
}
```

F#

```
member PageId : uint16 with get
```

Property Value

Type: [UInt16](#)

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.Version Property

Version of page being updated

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Version { get; }
```

VB

```
Public ReadOnly Property Version As ULong  
    Get
```

C++

```
public:  
property unsigned long long Version {  
    unsigned long long get ();  
}
```

F#

```
member Version : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also



[Change Class](#)

[VelocityDb.Sync Namespace](#)



Change.Change Methods

The [Change](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareTo(Object) | Compares objects using the object Id (Overrides OptimizedPersistable.CompareTo(Object) .) |
|  | CompareTo(Change) | Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |



See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.CompareTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareTo(Object) | Compares objects using the object Id (Overrides OptimizedPersistable.CompareTo(Object) .) |
|  | CompareTo(Change) | Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object. |

See Also

[Change Class](#)

[VelocityDb.Sync Namespace](#)

Change.CompareTo Method (Object)

Compares objects using the object [Id](#)

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int CompareTo(  
    Object change  
)
```

VB

```
Public Overrides Function CompareTo (  
    change As Object  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Object^ change  
) override
```

F#

```
abstract CompareTo :  
    change : Object -> int  
override CompareTo :  
    change : Object -> int
```

Parameters

change

Type: [System.Object](#)

[Missing <param name="change"/> documentation for "M:VelocityDb.Sync.Change.CompareTo(System.Object)"]

Return Value

Type: [Int32](#)

A value that indicates the relative order of the objects being compared. The return value has these meanings: Value Meaning Less than zero This instance precedes *obj* in the sort order. Zero This instance occurs in the same position in the sort order as *obj*. Greater than zero This instance follows *obj* in the sort order.

Implements

[IComparable.CompareTo\(Object\)](#)

[IComparable.CompareTo\(Object\)](#)

Exceptions

| Exception | Condition |
|-----------------------------------|------------------------------------------------------|
| ArgumentException | <i>obj</i> is not the same type as this instance. |

See Also

[Change Class](#)

[CompareTo Overload](#)

[VelocityDb.Sync Namespace](#)

Change.CompareTo Method (Change)

Compares the current instance with another object of the same type and returns an integer that indicates whether the current instance precedes, follows, or occurs in the same position in the sort order as the other object.

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    Change change  
)
```

VB

```
Public Function CompareTo (  
    change As Change  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Change^ change  
) sealed
```

F#

```
abstract CompareTo :  
    change : Change -> int  
override CompareTo :  
    change : Change -> int
```

Parameters

change

Type: [VelocityDb.Sync.Change](#)

[Missing <param name="change"/> documentation for "M:VelocityDb.Sync.Change.CompareTo(VelocityDb.Sync.Change)"]

Return Value

Type: [Int32](#)

A value that indicates the relative order of the objects being compared. The return value has these meanings: Value Meaning Less than zero This instance precedes *other* in the sort order. Zero This instance occurs in the same position in the sort order as *other*. Greater than zero This instance follows *other* in the sort order.

VelocityDB Class Library

Implements

[IComparable\(T\).CompareTo\(T\)](#)

See Also

[Change Class](#)

[CompareTo Overload](#)

[VelocityDb.Sync Namespace](#)

Changes Class

Keeps track of changes within a set of databases (a federation). Preliminary version.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Sync.Changes

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Changes : OptimizedPersistable
```

VB

```
Public Class Changes
    Inherits OptimizedPersistable
```

C++


```
public ref class Changes : public OptimizedPersistable
```

F#


```
type Changes =
    class
        inherit OptimizedPersistable
    end
```

The **Changes** type exposes the following members.



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------|
|  | ChangeList | A list of TransactionChanges made to the federation |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |


See Also

[VelocityDb.Sync Namespace](#)

Changes.Changes Properties

The [Changes](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------|
|  | ChangeList | A list of TransactionChanges made to the federation |

See Also

[Changes Class](#)

[VelocityDb.Sync Namespace](#)

Changes.ChangeList Property

A list of [TransactionChanges](#) made to the federation

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakReferenceList<TransactionChanges> ChangeList { get; }
```

VB

```
Public ReadOnly Property ChangeList As WeakReferenceList(Of  
TransactionChanges)  
    Get
```

C++

```
public:  
property WeakReferenceList<TransactionChanges^>^ ChangeList {  
    WeakReferenceList<TransactionChanges^>^ get ();  
}
```

F#

```
member ChangeList : WeakReferenceList<TransactionChanges> with get
```

Property Value

Type: [WeakReferenceList\(TransactionChanges\)](#)

See Also


[Changes Class](#)

[VelocityDb.Sync Namespace](#)



Changes.Changes Methods

The [Changes](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[Changes Class](#)

[VelocityDb.Sync Namespace](#)

Changes.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[Changes Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync Class

Contains info of latest transaction synced with databases at a specified host and path

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Sync.ReplicaSync

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReplicaSync : OptimizedPersistable
```

VB

```
Public Class ReplicaSync
    Inherits OptimizedPersistable
```

C++


```
public ref class ReplicaSync : public OptimizedPersistable
```

F#




```
type ReplicaSync =
    class
        inherit OptimizedPersistable
    end
```

The **ReplicaSync** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------------------|
|  | ReplicaSync | Creates sync log object |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-------------------------------------|
|  | SyncFromHost | Host of databases synced with |
|  | SyncFromPath | Path to databases synced with |
|  | TransactionNumber | Last transaction number synced with |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.Sync Namespace](#)

ReplicaSync Constructor

Creates sync log object

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReplicaSync(  
    SessionBase syncFromSession,  
    ulong transaction  
)
```

VB

```
Public Sub New (  
    syncFromSession As SessionBase,  
    transaction As ULong  
)
```

C++

```
public:  
ReplicaSync(  
    SessionBase^ syncFromSession,  
    unsigned long long transaction  
)
```

F#

```
new :  
    syncFromSession : SessionBase *  
    transaction : uint64 -> ReplicaSync
```

Parameters

syncFromSession

Type: [VelocityDb.Session.SessionBase](#)

The sync from session

transaction

Type: [System.UInt64](#)

Last transaction synced

See Also




[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync.ReplicaSync Properties

The [ReplicaSync](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------------------------------|
|  | SyncFromHost | Host of databases synced with |
|  | SyncFromPath | Path to databases synced with |
|  | TransactionNumber | Last transaction number synced with |

See Also

[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync.SyncFromHost Property

Host of databases synced with

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SyncFromHost { get; }
```

VB

```
Public ReadOnly Property SyncFromHost As String  
    Get
```

C++

```
public:  
property String^ SyncFromHost {  
    String^ get ();  
}
```

F#

```
member SyncFromHost : string with get
```

Property Value

Type: [String](#)

See Also

[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync.SyncFromPath Property

Path to databases synced with

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SyncFromPath { get; }
```

VB

```
Public ReadOnly Property SyncFromPath As String  
    Get
```

C++

```
public:  
property String^ SyncFromPath {  
    String^ get ();  
}
```

F#

```
member SyncFromPath : string with get
```

Property Value

Type: [String](#)

See Also

[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync.TransactionNumber Property

Last transaction number synced with

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong TransactionNumber { get; set; }
```

VB

```
Public Property TransactionNumber As ULong  
    Get  
    Set
```

C++

```
public:  
property unsigned long long TransactionNumber {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
member TransactionNumber : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also



[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

ReplicaSync.ReplicaSync Methods

The [ReplicaSync](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[ReplicaSync Class](#)

[VelocityDb.Sync Namespace](#)

TransactionChanges Class

A log of all changes made within a certain transaction.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.Sync.TransactionChanges

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class TransactionChanges : OptimizedPersistable
```

VB

```
Public Class TransactionChanges
    Inherits OptimizedPersistable
```

C++



```
public ref class TransactionChanges : public OptimizedPersistable
```

F#


```
type TransactionChanges =
    class
        inherit OptimizedPersistable
    end
```

The **TransactionChanges** type exposes the following members.



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-----------------------------------|
|  | ChangeList | List of changes made |
|  | TransactionNumber | Transaction id for these changes. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |



See Also

[VelocityDb.Sync Namespace](#)

TransactionChanges.TransactionChanges Properties

The [TransactionChanges](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-----------------------------------|
|  | ChangeList | List of changes made |
|  | TransactionNumber | Transaction id for these changes. |

See Also

[TransactionChanges Class](#)

[VelocityDb.Sync Namespace](#)

TransactionChanges.ChangeList Property

List of changes made

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WeakReferenceList<Change> ChangeList { get; }
```

VB

```
Public ReadOnly Property ChangeList As WeakReferenceList(Of Change)  
    Get
```

C++

```
public:  
property WeakReferenceList<Change^>^ ChangeList {  
    WeakReferenceList<Change^>^ get ();  
}
```

F#

```
member ChangeList : WeakReferenceList<Change> with get
```

Property Value

Type: [WeakReferenceList\(Change\)](#)

See Also

[TransactionChanges Class](#)

[VelocityDb.Sync Namespace](#)

TransactionChanges.TransactionNumber Property

Transaction id for these changes.

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong TransactionNumber { get; }
```

VB

```
Public ReadOnly Property TransactionNumber As ULong  
    Get
```

C++

```
public:  
property unsigned long long TransactionNumber {  
    unsigned long long get ();  
}
```

F#

```
member TransactionNumber : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also


[TransactionChanges Class](#)

[VelocityDb.Sync Namespace](#)



TransactionChanges.TransactionChanges Methods

The [TransactionChanges](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[TransactionChanges Class](#)

[VelocityDb.Sync Namespace](#)

TransactionChanges.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.Sync](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also
















[TransactionChanges Class](#)

[VelocityDb.Sync Namespace](#)


VelocityDb.TypeInfo Namespace

The `VelocityDb.TypeInfo` namespace contains classes for Type information of persistently stored data

Classes

| Class | Description |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Association | A field attribute that controls what to do when deleting object and other end of association exist. |
|  AutoIncrement | A field attribute that enables a field to get a unique number for each persisted object of a certain type. This only works for objects persisted (directly or indirectly) with Persist(IOptimizedPersistable, Nullable(UInt16)) |
|  CommonTypes | A cache of commonly instantiated Types and their size in bytes |
|  DataMember | Info about a data Field of a persistent object |
|  FieldAccessor | A field attribute that controls how many objects that will be placed on the same page as the object of the field. |
|  ObjectsPerPage | A field attribute that controls how many objects that will be placed on the same page as the object of the field. |
|  Reference | |
|  Relation | The relation classes exist as an aid to maintain referential integrity, that is no dangling references (references to deleted objects). NOTE: These classes may change as they are still in a prototype phase. Let us know how we can improve them? Any ideas are appreciated! |
|  RelationManyToMany(From, To) | Use for many to many relations |
|  RelationOneToMany(From, To) | Use for one to many (and many to one) relations. |
|  RelationOneToOne(From, To) | Use for one to one relations. |
|  Schema | Holds information about persistently stored types. Used internally by VelocityDb. |
|  TypeVersion | Contains info about a version of a VelocityDbType . The info is used when reading/writing an object of the |
|  UseOidShort | Indicates that every field or selected fields of a serializable class should be referenced using a OidShort instead of a full Oid. This means that the referenced object is located in the same database as the object containing the reference. A ShortOid uses less space and references are not tied to a certain database number. |
|  VelocityDbType | Contains persistent info for a Type |

Enumerations

| | Enumeration | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------|------------------------------------------------------|
|  | Association.ReverseReferenceTypeEnum | What type of reference is used by reverse reference? |

Association Class

A field attribute that controls what to do when deleting object and other end of association exist.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.TypeInfo.Association

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class Association : Attribute
```

VB

```
Public NotInheritable Class Association
    Inherits Attribute
```

C++



```
public ref class Association sealed : public Attribute
```

F#



```
[<SealedAttribute>]
type Association =
    class
        inherit Attribute
    end
```

The **Association** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------|
|  | Association(String) | Initializes a new instance of the Association class |
|  | Association(String, Association.ReverseReferenceTypeEnum) | Initializes a new instance of the Association class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------|
|  | ReversePropertyName | Name of property that retrieves other side of association |
|  | ReverseReferenceType | Type of reference to this object at other end of association |



VelocityDB Class Library

See Also

[VelocityDb.TypeInfo Namespace](#)

Association Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------|
|  | Association(String) | Initializes a new instance of the Association class |
|  | Association(String, Association.ReverseReferenceTypeEnum) | Initializes a new instance of the Association class |

See Also

[Association Class](#)

[VelocityDb.TypeInfo Namespace](#)

Association Constructor (String)

Initializes a new instance of the [Association](#) class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Association(  
    string reversePropertyName  
)
```

VB

```
Public Sub New (  
    reversePropertyName As String  
)
```

C++

```
public:  
Association(  
    String^ reversePropertyName  
)
```

F#

```
new :  
    reversePropertyName : string -> Association
```

Parameters

reversePropertyName

Type: [System.String](#)

[Missing <param name="reversePropertyName"/> documentation for "M:VelocityDb.TypeInfo.Association.#ctor(System.String)"]

See Also

[Association Class](#)

[Association Overload](#)

[VelocityDb.TypeInfo Namespace](#)

Association Constructor (String, Association.ReverseReferenceTypeEnum)

Initializes a new instance of the [Association](#) class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Association(  
    string reversePropertyName,  
    Association.ReverseReferenceTypeEnum reverseReferenceType  
)
```

VB

```
Public Sub New (  
    reversePropertyName As String,  
    reverseReferenceType As Association.ReverseReferenceTypeEnum  
)
```

C++

```
public:  
Association(  
    String^ reversePropertyName,  
    Association.ReverseReferenceTypeEnum reverseReferenceType  
)
```

F#

```
new :  
    reversePropertyName : string *  
    reverseReferenceType : Association.ReverseReferenceTypeEnum ->  
Association
```

Parameters

reversePropertyName

Type: [System.String](#)

[Missing <param name="reversePropertyName"/> documentation for "M:VelocityDb.TypeInfo.Association.#ctor(System.String,VelocityDb.TypeInfo.Association.ReverseReferenceTypeEnum)"]

reverseReferenceType

Type: [VelocityDb.TypeInfo.Association.ReverseReferenceTypeEnum](#)

[Missing <param name="reverseReferenceType"/> documentation for "M:VelocityDb.TypeInfo.Association.#ctor(System.String,VelocityDb.TypeInfo.Association.ReverseReferenceTypeEnum)"]

VelocityDB Class Library

See Also

[Association Class](#)



[Association Overload](#)

[VelocityDb.TypeInfo Namespace](#)

Association.Association Properties

The [Association](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------|
|  | ReversePropertyName | Name of property that retrieves other side of association |
|  | ReverseReferenceType | Type of reference to this object at other end of association |

See Also

[Association Class](#)

[VelocityDb.TypeInfo Namespace](#)

Association.ReversePropertyName Property

Name of property that retrieves other side of association

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string ReversePropertyName { get; }
```

VB

```
Public ReadOnly Property ReversePropertyName As String  
    Get
```

C++

```
public:  
property String^ ReversePropertyName {  
    String^ get ();  
}
```

F#

```
member ReversePropertyName : string with get
```

Property Value

Type: [String](#)

See Also

[Association Class](#)

[VelocityDb.TypeInfo Namespace](#)

Association.ReverseReferenceType Property

Type of reference to this object at other end of association

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Association.ReverseReferenceTypeEnum ReverseReferenceType { get; }
```

VB

```
Public ReadOnly Property ReverseReferenceType As  
Association.ReverseReferenceTypeEnum  
Get
```

C++

```
public:  
property Association.ReverseReferenceTypeEnum ReverseReferenceType {  
    Association.ReverseReferenceTypeEnum get ();  
}
```

F#

```
member ReverseReferenceType : Association.ReverseReferenceTypeEnum with get
```

Property Value

Type: [Association.ReverseReferenceTypeEnum](#)

See Also

[Association Class](#)

[VelocityDb.TypeInfo Namespace](#)

Association.ReverseReferenceTypeEnum Enumeration

What type of reference is used by reverse reference?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum ReverseReferenceTypeEnum
```

VB

```
Public Enumeration ReverseReferenceTypeEnum
```

C++

```
public enum class ReverseReferenceTypeEnum
```

F#

```
type ReverseReferenceTypeEnum
```

Members

| Member name | Value | Description |
|-------------|-------|------------------------------------------------|
| ToOne | 0 | A direct reference. |
| InSet | 1 | Reference is within a ISet(T) |
| InList | 2 | Reference is within a IList(T) |
| InArray | 3 | Reference is within a Array |

See Also

[VelocityDb.TypeInfo Namespace](#)

AutoIncrement Class

A field attribute that enables a field to get a unique number for each persisted object of a certain type. This only works for objects persisted (directly or indirectly) with [Persist\(IOptimizedPersistable, Nullable\(UInt16\)\)](#)

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.TypeInfo.AutoIncrement

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class AutoIncrement : Attribute
```

VB

```
Public NotInheritable Class AutoIncrement
    Inherits Attribute
```

C++


```
public ref class AutoIncrement sealed : public Attribute
```

F#


```
[<SealedAttribute>]
type AutoIncrement =
    class
        inherit Attribute
    end
```

The **AutoIncrement** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------------------------|
|  | AutoIncrement | Use in class definition preceding a field declaration, i.e. [AutoIncrement(1)] |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|----------------------------------------------------------------|
|  | Start | Gets the requested start number of the auto increment sequence |

VelocityDB Class Library

See Also

[VelocityDb.TypeInfo Namespace](#)

AutoIncrement Constructor

Use in class definition preceding a field declaration, i.e. [AutoIncrement(1)]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public AutoIncrement(  
    ushort start = 0  
)
```

VB

```
Public Sub New (  
    Optional start As UShort = 0  
)
```

C++

```
public:  
AutoIncrement(  
    unsigned short start = 0  
)
```

F#

```
new :  
    ?start : uint16  
(* Defaults:  
    let _start = defaultArg start 0  
)  
-> AutoIncrement
```

Parameters

start (Optional)

Type: [System.UInt16](#)

The requested start number of the auto increment sequence. Default is 0 so that first persisted object is assigned number 1, second number 2 and so on.

See Also


[AutoIncrement Class](#)

[VelocityDb.TypeInfo Namespace](#)

AutoIncrement.AutoIncrement Properties

The [AutoIncrement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|----------------------------------------------------------------|
|  | Start | Gets the requested start number of the auto increment sequence |

See Also

[AutoIncrement Class](#)

[VelocityDb.TypeInfo Namespace](#)

AutoIncrement.Start Property

Gets the requested start number of the auto increment sequence

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong Start { get; }
```

VB

```
Public ReadOnly Property Start As ULong  
    Get
```

C++

```
public:  
property unsigned long long Start {  
    unsigned long long get ();  
}
```

F#

```
member Start : uint64 with get
```

Property Value

Type: [UInt64](#)

The [UInt64](#) sequence start number.

See Also

[AutoIncrement Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes Class

A cache of commonly instantiated [Types](#) and their size in bytes

Inheritance Hierarchy

[System.Object](#)

VelocityDb.TypeInfo.CommonTypes

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class CommonTypes
```

VB

```
Public NotInheritable Class CommonTypes
```

C++













```
public ref class CommonTypes abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type CommonTypes = class end
```

The **CommonTypes** type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|--------------------------------------------|
|   | s_sizeOfBoolean | Cache of sizeof(Boolean) |
|   | s_sizeOfByte | Cache of sizeof(Byte) |
|   | s_sizeOfChar | Cache of sizeof(Char) |
|   | s_sizeOfDecimal | Cache of sizeof(Decimal) |
|   | s_sizeOfDouble | Cache of sizeof(Double) |
|   | s_sizeOfInt16 | Cache of sizeof(Int16) |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
|  s_sizeOfInt32  | Cache of sizeof(Int32) |
|  s_sizeOfInt64  | Cache of sizeof(Int64) |
|  s_sizeOfSByte  | Cache of sizeof(SByte) |
|  s_sizeOfSingle  | Cache of sizeof(Single) |
|  s_sizeOfUInt16  | Cache of sizeof(UInt16) |
|  s_sizeOfUInt32  | Cache of sizeof(UInt32) |
|  s_sizeOfUInt64  | Cache of sizeof(UInt64) |
|  s_typeOfArray  | Cache of typeof(Array) |
|  s_typeOfArrayList  | Cache of typeof(ArrayList) |
|  s_typeOfBoolean  | Cache of typeof(Boolean) |
|  s_typeOfBTreeInternal  | Cache of typeof(BTreeInternal(Key, Value)) |
|  s_typeOfBTreeInternalOidShort  | Cache of typeof(BTreeInternalOidShort) |
|  s_typeOfBTreeLeaf  | Cache of typeof(BTreeLeaf(Key, Value)) |
|  s_typeOfBTreeLeafBase  | Cache of typeof(BTreeLeafBase(Key, Value)) |
|  s_typeOfBTreeLeafOidShort  | Cache of typeof(BTreeLeafOidShort) |
|  s_typeOfBTreeMap  | Cache of typeof(BTreeMap(Key, Value)) |
|  s_typeOfBTreeMapOidShort  | Cache of typeof(BTreeMapOidShort(Key, Value)) |
|  s_typeOfBTreeSet  | Cache of typeof(BTreeSet(Key)) |
|  s_typeOfBTreeSetOidShort  | Cache of typeof(BTreeSetOidShort(Key)) |
|  s_typeOfByte  | Cache of typeof(Byte) |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
|  s_typeofByteArray  | Cache of typeof(Byte[]) |
|  s_typeofChar  | Cache of typeof(Char) |
|  s_typeofCompareByFieldIndex  | Cache of typeof(CompareByFieldIndex(Key)) |
|  s_typeofDataForISerializable  | Cache of typeof(DataForISerializable) |
|  s_typeofDateTime  | Cache of typeof(DateTime) |
|  s_typeofDecimal  | Cache of typeof(Decimal) |
|  s_typeofDelegate  | Cache of typeof(Delegate) |
|  s_typeofDouble  | Cache of typeof(Double) |
|  s_typeofEmbed  | Cache of typeof(Embed) |
|  s_typeofFunc  | Cache of typeof(Func{}) |
|  s_typeofGetHandler  | Cache of typeof(Func(object, object);) |
|  s_typeofGuid  | Cache of typeof(Guid) |
|  s_typeofIndex  | Cache of typeof(Index) |
|  s_typeofIndexStringByHashCode  | Cache of typeof(IndexStringByHashCode) |
|  s_typeofInt16  | Cache of typeof(Int16) |
|  s_typeofInt32  | Cache of typeof(Int32) |
|  s_typeofInt64  | Cache of typeof(Int64) |
|  s_typeofIOptimizedPersistable  | Cache of typeof(IOptimizedPersistable) |
|  s_typeofISet  | Cache of typeof(ISet(T)) |
|  s_typeofList  | Cache of typeof(List{}) |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|  s_typeOfLong  | Cache of typeof(Int64) |
|  s_typeOfNonSerializedAttribute  | Cache of typeof(NonSerializedAttribute) |
|  s_typeOfNullable  | Cache of typeof(Nullable) |
|  s_typeOfObject  | Cache of typeof(Object) |
|  s_typeOfObjectActivator  | Cache of typeof(ObjectActivator) |
|  s_typeOfObjectArray  | Cache of typeof(Object []) |
|  s_typeOfOid  | Cache of typeof(Oid) |
|  s_typeOfOidShort  | Cache of typeof(OidShort) |
|  s_typeOfOnePerDatabase  | Cache of typeof(OnePerDatabase) |
|  s_typeOfOptimizedPersistable  | Cache of typeof(OptimizedPersistable) |
|  s_typeOfSByte  | Cache of typeof(SByte) |
|  s_typeOfSchema  | Cache of typeof(Schema) |
|  s_typeOfSessionBase  | Cache of typeof(SessionBase) |
|  s_typeOfSetHandler  | Cache of typeof(Action {object, object}); |
|  s_typeOfSingle  | Cache of typeof(Single) |
|  s_typeOfString  | Cache of typeof(String) |
|  s_typeOfStringArray  | Cache of typeof(String []) |
|  s_typeOfTimeSpan  | Cache of typeof(TimeSpan) |
|  s_typeOfType  | Cache of typeof(Type) |
|  s_typeOfTypeArray  | Cache of typeof(Type []) |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|  s_typeofTypeVersionArray  | Cache of typeof(TypeVersion[]) |
|  s_typeofUInt16  | Cache of typeof(UInt16) |
|  s_typeofUInt32  | Cache of typeof(UInt32) |
|  s_typeofUInt64  | Cache of typeof(UInt64) |
|  s_typeofUniqueConstraint  | Cache of typeof(UniqueConstraint) |
|  s_typeofUseOidShort  | Cache of typeof(UseOidShort) |
|  s_typeofValueType  | Cache of typeof(ValueType) |
|  s_typeofVelocityDbComparer  | Cache of typeof(VelocityDbComparer(Key)) |
|  s_typeofVelocityDbList  | Cache of typeof(VelocityDbList(T)) |
|  s_typeofVelocityDbListOidShort  | Cache of typeof(VelocityDbListOidShort(T)) |
|  s_typeofWeakIOptimizedPersistableReference  | Cache of typeof(WeakIOptimizedPersistableReference) |
|  s_typeofWeakIOptimizedPersistableReferenceBase  | Cache of typeof(WeakIOptimizedPersistableReferenceBase) |
|  s_typeofWeakReferenceList  | Cache of typeof(WeakReferenceList(T)) |
|  s_typeofWeakShortReferenceList  | Cache of typeof(WeakShortReferenceList(T)) |

See Also







[VelocityDb.TypeInfo Namespace](#)

CommonTypes.CommonTypes Fields

The [CommonTypes](#) type exposes the following members.

Fields







| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|  s_sizeOfBoolean  | Cache of sizeof(Boolean) |
|  s_sizeOfByte  | Cache of sizeof(Byte) |
|  s_sizeOfChar  | Cache of sizeof(Char) |
|  s_sizeOfDecimal  | Cache of sizeof(Decimal) |
|  s_sizeOfDouble  | Cache of sizeof(Double) |
|  s_sizeOfInt16  | Cache of sizeof(Int16) |
|  s_sizeOfInt32  | Cache of sizeof(Int32) |
|  s_sizeOfInt64  | Cache of sizeof(Int64) |
|  s_sizeOfSByte  | Cache of sizeof(SByte) |
|  s_sizeOfSingle  | Cache of sizeof(Single) |
|  s_sizeOfUInt16  | Cache of sizeof(UInt16) |
|  s_sizeOfUInt32  | Cache of sizeof(UInt32) |
|  s_sizeOfUInt64  | Cache of sizeof(UInt64) |
|  s_typeofArray  | Cache of typeof(Array) |
|  s_typeofArrayList  | Cache of typeof(ArrayList) |
|  s_typeofBoolean  | Cache of typeof(Boolean) |
|  s_typeofBTreeInternal  | Cache of typeof(BTreeInternal(Key, Value)) |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
|  s_typeOfBTreeInternalOidShort  | Cache of typeof(BTreeInternalOidShort) |
|  s_typeOfBTreeLeaf  | Cache of typeof(BTreeLeaf(Key, Value)) |
|  s_typeOfBTreeLeafBase  | Cache of typeof(BTreeLeafBase(Key, Value)) |
|  s_typeOfBTreeLeafOidShort  | Cache of typeof(BTreeLeafOidShort) |
|  s_typeOfBTreeMap  | Cache of typeof(BTreeMap(Key, Value)) |
|  s_typeOfBTreeMapOidShort  | Cache of typeof(BTreeMapOidShort(Key, Value)) |
|  s_typeOfBTreeSet  | Cache of typeof(BTreeSet(Key)) |
|  s_typeOfBTreeSetOidShort  | Cache of typeof(BTreeSetOidShort(Key)) |
|  s_typeOfByte  | Cache of typeof(Byte) |
|  s_typeOfByteArray  | Cache of typeof(Byte[]) |
|  s_typeOfChar  | Cache of typeof(Char) |
|  s_typeOfCompareByFieldIndex  | Cache of typeof(CompareByFieldIndex(Key)) |
|  s_typeOfDataForISerializable  | Cache of typeof(DataForISerializable) |
|  s_typeOfDateTime  | Cache of typeof(DateTime) |
|  s_typeOfDecimal  | Cache of typeof(Decimal) |
|  s_typeOfDelegate  | Cache of typeof(Delegate) |
|  s_typeOfDouble  | Cache of typeof(Double) |
|  s_typeOfEmbed  | Cache of typeof(Embed) |
|  s_typeOfFunc  | Cache of typeof(Func{}) |
|  s_typeOfGetHandler  | Cache of typeof(Func{object, object}); |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|
|  s_typeofGuid  | Cache of typeof(Guid) |
|  s_typeofIndex  | Cache of typeof(Index) |
|  s_typeofIndexStringByHashCode  | Cache of typeof(IndexStringByHashCode) |
|  s_typeofInt16  | Cache of typeof(Int16) |
|  s_typeofInt32  | Cache of typeof(Int32) |
|  s_typeofInt64  | Cache of typeof(Int64) |
|  s_typeofIOptimizedPersistable  | Cache of typeof(IOptimizedPersistable) |
|  s_typeofISet  | Cache of typeof(ISet(T)) |
|  s_typeofList  | Cache of typeof(List{}) |
|  s_typeofLong  | Cache of typeof(Int64) |
|  s_typeofNonSerializedAttribute  | Cache of typeof(NonSerializedAttribute) |
|  s_typeofNullable  | Cache of typeof(Nullable) |
|  s_typeofObject  | Cache of typeof(Object) |
|  s_typeofObjectActivator  | Cache of typeof(ObjectActivator) |
|  s_typeofObjectArray  | Cache of typeof(Object[]) |
|  s_typeofOid  | Cache of typeof(Oid) |
|  s_typeofOidShort  | Cache of typeof(OidShort) |
|  s_typeofOnePerDatabase  | Cache of typeof(OnePerDatabase) |
|  s_typeofOptimizedPersistable  | Cache of typeof(OptimizedPersistable) |
|  s_typeofSByte  | Cache of typeof(SByte) |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
|  s_typeOfSchema  | Cache of typeof(Schema) |
|  s_typeOfSessionBase  | Cache of typeof(SessionBase) |
|  s_typeOfSetHandler  | Cache of typeof(Action{object, object}); |
|  s_typeOfSingle  | Cache of typeof(Single) |
|  s_typeOfString  | Cache of typeof(String) |
|  s_typeOfStringArray  | Cache of typeof(String[]) |
|  s_typeOfTimeSpan  | Cache of typeof(TimeSpan) |
|  s_typeOfType  | Cache of typeof(Type) |
|  s_typeOfTypeArray  | Cache of typeof(Type[]) |
|  s_typeOfTypeVersionArray  | Cache of typeof(TypeVersion[]) |
|  s_typeOfUInt16  | Cache of typeof(UInt16) |
|  s_typeOfUInt32  | Cache of typeof(UInt32) |
|  s_typeOfUInt64  | Cache of typeof(UInt64) |
|  s_typeOfUniqueConstraint  | Cache of typeof(UniqueConstraint) |
|  s_typeOfUseOidShort  | Cache of typeof(UseOidShort) |
|  s_typeOfValueType  | Cache of typeof(ValueType) |
|  s_typeOfVelocityDbComparer  | Cache of typeof(VelocityDbComparer(Key)) |
|  s_typeOfVelocityDbList  | Cache of typeof(VelocityDbList(T)) |
|  s_typeOfVelocityDbListOidShort  | Cache of typeof(VelocityDbListOidShort(T)) |
|  s_typeOfWeakIOptimizedPersistableReference  | Cache of typeof(WeakIOptimizedPersistableReference) |

VelocityDB Class Library

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
|  s_typeofWeakIOptimizedPersistableReferenceBase  se | Cache of typeof(WeakIOptimizedPersistableReferenceBase) |
|  s_typeofWeakReferenceList  | Cache of typeof(WeakReferenceList(T)) |
|  s_typeofWeakShortReferenceList  | Cache of typeof(WeakShortReferenceList(T)) |

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfBoolean Field

Cache of sizeof([Boolean](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfBoolean
```

VB

```
Public Shared ReadOnly s_sizeOfBoolean As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfBoolean
```

F#

```
static val s_sizeOfBoolean: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfByte Field

Cache of sizeof([Byte](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfByte
```

VB

```
Public Shared ReadOnly s_sizeOfByte As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfByte
```

F#

```
static val s_sizeOfByte: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfChar Field

Cache of sizeof([Char](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfChar
```

VB

```
Public Shared ReadOnly s_sizeOfChar As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfChar
```

F#

```
static val s_sizeOfChar: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfDecimal Field

Cache of sizeof([Decimal](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfDecimal
```

VB

```
Public Shared ReadOnly s_sizeOfDecimal As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfDecimal
```

F#

```
static val s_sizeOfDecimal: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfDouble Field

Cache of sizeof([Double](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfDouble
```

VB

```
Public Shared ReadOnly s_sizeOfDouble As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfDouble
```

F#

```
static val s_sizeOfDouble: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfInt16 Field

Cache of sizeof([Int16](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfInt16
```

VB

```
Public Shared ReadOnly s_sizeOfInt16 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfInt16
```

F#

```
static val s_sizeOfInt16: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfInt32 Field

Cache of sizeof([Int32](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfInt32
```

VB

```
Public Shared ReadOnly s_sizeOfInt32 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfInt32
```

F#

```
static val s_sizeOfInt32: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfInt64 Field

Cache of sizeof([Int64](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfInt64
```

VB

```
Public Shared ReadOnly s_sizeOfInt64 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfInt64
```

F#

```
static val s_sizeOfInt64: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfSByte Field

Cache of sizeof([SByte](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfSByte
```

VB

```
Public Shared ReadOnly s_sizeOfSByte As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfSByte
```

F#

```
static val s_sizeOfSByte: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfSingle Field

Cache of sizeof([Single](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfSingle
```

VB

```
Public Shared ReadOnly s_sizeOfSingle As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfSingle
```

F#

```
static val s_sizeOfSingle: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfUInt16 Field

Cache of sizeof([UInt16](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfUInt16
```

VB

```
Public Shared ReadOnly s_sizeOfUInt16 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfUInt16
```

F#

```
static val s_sizeOfUInt16: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfUInt32 Field

Cache of sizeof([UInt32](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfUInt32
```

VB

```
Public Shared ReadOnly s_sizeOfUInt32 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfUInt32
```

F#

```
static val s_sizeOfUInt32: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_sizeOfUInt64 Field

Cache of sizeof([UInt64](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_sizeOfUInt64
```

VB

```
Public Shared ReadOnly s_sizeOfUInt64 As UShort
```

C++

```
public:  
static initonly unsigned short s_sizeOfUInt64
```

F#

```
static val s_sizeOfUInt64: uint16
```

Field Value

Type: [UInt16](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofArray Field

Cache of typeof([Array](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofArray
```

VB

```
Public Shared ReadOnly s_typeofArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofArray
```

F#

```
static val s_typeofArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofArrayList Field

Cache of typeof(ArrayList)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofArrayList
```

VB

```
Public Shared ReadOnly s_typeofArrayList As Type
```

C++

```
public:  
static initonly Type^ s_typeofArrayList
```

F#

```
static val s_typeofArrayList: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBoolean Field

Cache of typeof([Boolean](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBoolean
```

VB

```
Public Shared ReadOnly s_typeofBoolean As Type
```

C++

```
public:  
static initonly Type^ s_typeofBoolean
```

F#

```
static val s_typeofBoolean: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeInternal Field

Cache of typeof([BTreeInternal\(Key, Value\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeInternal
```

VB

```
Public Shared ReadOnly s_typeofBTreeInternal As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeInternal
```

F#

```
static val s_typeofBTreeInternal: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeInternalOidShort Field

Cache of typeof(**BTreeInternalOidShort**)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeInternalOidShort
```

VB

```
Public Shared ReadOnly s_typeofBTreeInternalOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeInternalOidShort
```

F#

```
static val s_typeofBTreeInternalOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeLeaf Field

Cache of typeof([BTreeLeaf\(Key, Value\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeLeaf
```

VB

```
Public Shared ReadOnly s_typeofBTreeLeaf As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeLeaf
```

F#

```
static val s_typeofBTreeLeaf: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeLeafBase Field

Cache of typeof([BTreeLeafBase\(Key, Value\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeLeafBase
```

VB

```
Public Shared ReadOnly s_typeofBTreeLeafBase As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeLeafBase
```

F#

```
static val s_typeofBTreeLeafBase: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeLeafOidShort Field

Cache of typeof(**BTreeLeafOidShort**)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeLeafOidShort
```

VB

```
Public Shared ReadOnly s_typeofBTreeLeafOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeLeafOidShort
```

F#

```
static val s_typeofBTreeLeafOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeMap Field

Cache of typeof([BTreeMap\(Key, Value\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeMap
```

VB

```
Public Shared ReadOnly s_typeofBTreeMap As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeMap
```

F#

```
static val s_typeofBTreeMap: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeMapOidShort Field

Cache of typeof([BTreeMapOidShort\(Key, Value\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeMapOidShort
```

VB

```
Public Shared ReadOnly s_typeofBTreeMapOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeMapOidShort
```

F#

```
static val s_typeofBTreeMapOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeOfBTreeSet Field

Cache of typeof([BTreeSet\(Key\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeOfBTreeSet
```

VB

```
Public Shared ReadOnly s_typeOfBTreeSet As Type
```

C++

```
public:  
static initonly Type^ s_typeOfBTreeSet
```

F#

```
static val s_typeOfBTreeSet: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofBTreeSetOidShort Field

Cache of typeof([BTreeSetOidShort\(Key\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofBTreeSetOidShort
```

VB

```
Public Shared ReadOnly s_typeofBTreeSetOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofBTreeSetOidShort
```

F#

```
static val s_typeofBTreeSetOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofByte Field

Cache of typeof([Byte](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofByte
```

VB

```
Public Shared ReadOnly s_typeofByte As Type
```

C++

```
public:  
static initonly Type^ s_typeofByte
```

F#

```
static val s_typeofByte: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofByteArray Field

Cache of typeof([Byte\[\]](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofByteArray
```

VB

```
Public Shared ReadOnly s_typeofByteArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofByteArray
```

F#

```
static val s_typeofByteArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofChar Field

Cache of typeof([Char](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofChar
```

VB

```
Public Shared ReadOnly s_typeofChar As Type
```

C++

```
public:  
static initonly Type^ s_typeofChar
```

F#

```
static val s_typeofChar: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofCompareByFieldIndex Field

Cache of typeof([CompareByFieldIndex\(Key\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofCompareByFieldIndex
```

VB

```
Public Shared ReadOnly s_typeofCompareByFieldIndex As Type
```

C++

```
public:  
static initonly Type^ s_typeofCompareByFieldIndex
```

F#

```
static val s_typeofCompareByFieldIndex: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofDataForISerializable Field

Cache of typeof(**DataForISerializable**)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofDataForISerializable
```

VB

```
Public Shared ReadOnly s_typeofDataForISerializable As Type
```

C++

```
public:  
static initonly Type^ s_typeofDataForISerializable
```

F#

```
static val s_typeofDataForISerializable: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofDateTime Field

Cache of typeof([DateTime](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofDateTime
```

VB

```
Public Shared ReadOnly s_typeofDateTime As Type
```

C++

```
public:  
static initonly Type^ s_typeofDateTime
```

F#

```
static val s_typeofDateTime: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofDecimal Field

Cache of typeof([Decimal](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofDecimal
```

VB

```
Public Shared ReadOnly s_typeofDecimal As Type
```

C++

```
public:  
static initonly Type^ s_typeofDecimal
```

F#

```
static val s_typeofDecimal: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofDelegate Field

Cache of typeof(Delegate)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofDelegate
```

VB

```
Public Shared ReadOnly s_typeofDelegate As Type
```

C++

```
public:  
static initonly Type^ s_typeofDelegate
```

F#

```
static val s_typeofDelegate: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofDouble Field

Cache of typeof([Double](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofDouble
```

VB

```
Public Shared ReadOnly s_typeofDouble As Type
```

C++

```
public:  
static initonly Type^ s_typeofDouble
```

F#

```
static val s_typeofDouble: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofEmbed Field

Cache of typeof(Embed)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofEmbed
```

VB

```
Public Shared ReadOnly s_typeofEmbed As Type
```

C++

```
public:  
static initonly Type^ s_typeofEmbed
```

F#

```
static val s_typeofEmbed: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofFunc Field

Cache of typeof(Func{ })

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofFunc
```

VB

```
Public Shared ReadOnly s_typeofFunc As Type
```

C++

```
public:  
static initonly Type^ s_typeofFunc
```

F#

```
static val s_typeofFunc: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofHandler Field

Cache of typeof(Func{object, object});

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofHandler
```

VB

```
Public Shared ReadOnly s_typeofHandler As Type
```

C++

```
public:  
static initonly Type^ s_typeofHandler
```

F#

```
static val s_typeofHandler: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofGuid Field

Cache of typeof([Guid](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofGuid
```

VB

```
Public Shared ReadOnly s_typeofGuid As Type
```

C++

```
public:  
static initonly Type^ s_typeofGuid
```

F#

```
static val s_typeofGuid: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofIndex Field

Cache of typeof([Index](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofIndex
```

VB

```
Public Shared ReadOnly s_typeofIndex As Type
```

C++

```
public:  
static initonly Type^ s_typeofIndex
```

F#

```
static val s_typeofIndex: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofIndexStringByHashCode Field

Cache of typeof(IndexStringByHashCode)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofIndexStringByHashCode
```

VB

```
Public Shared ReadOnly s_typeofIndexStringByHashCode As Type
```

C++

```
public:  
static initonly Type^ s_typeofIndexStringByHashCode
```

F#

```
static val s_typeofIndexStringByHashCode: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofInt16 Field

Cache of typeof([Int16](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofInt16
```

VB

```
Public Shared ReadOnly s_typeofInt16 As Type
```

C++

```
public:  
static initonly Type^ s_typeofInt16
```

F#

```
static val s_typeofInt16: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofInt32 Field

Cache of typeof([Int32](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofInt32
```

VB

```
Public Shared ReadOnly s_typeofInt32 As Type
```

C++

```
public:  
static initonly Type^ s_typeofInt32
```

F#

```
static val s_typeofInt32: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofInt64 Field

Cache of typeof([Int64](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofInt64
```

VB

```
Public Shared ReadOnly s_typeofInt64 As Type
```

C++

```
public:  
static initonly Type^ s_typeofInt64
```

F#

```
static val s_typeofInt64: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofIOptimizedPersistable Field

Cache of typeof([IOptimizedPersistable](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofIOptimizedPersistable
```

VB

```
Public Shared ReadOnly s_typeofIOptimizedPersistable As Type
```

C++

```
public:  
static initonly Type^ s_typeofIOptimizedPersistable
```

F#

```
static val s_typeofIOptimizedPersistable: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofISet Field

Cache of typeof([ISet\(T\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofISet
```

VB

```
Public Shared ReadOnly s_typeofISet As Type
```

C++

```
public:  
static initonly Type^ s_typeofISet
```

F#

```
static val s_typeofISet: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofList Field

Cache of typeof(List{ })

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofList
```

VB

```
Public Shared ReadOnly s_typeofList As Type
```

C++

```
public:  
static initonly Type^ s_typeofList
```

F#

```
static val s_typeofList: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofLong Field

Cache of typeof([Int64](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofLong
```

VB

```
Public Shared ReadOnly s_typeofLong As Type
```

C++

```
public:  
static initonly Type^ s_typeofLong
```

F#

```
static val s_typeofLong: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofNonSerializedAttribute Field

Cache of typeof(NonSerializedAttribute)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofNonSerializedAttribute
```

VB

```
Public Shared ReadOnly s_typeofNonSerializedAttribute As Type
```

C++

```
public:  
static initonly Type^ s_typeofNonSerializedAttribute
```

F#

```
static val s_typeofNonSerializedAttribute: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofNullable Field

Cache of typeof(Nullable)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofNullable
```

VB

```
Public Shared ReadOnly s_typeofNullable As Type
```

C++

```
public:  
static initonly Type^ s_typeofNullable
```

F#

```
static val s_typeofNullable: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofObject Field

Cache of typeof([Object](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofObject
```

VB

```
Public Shared ReadOnly s_typeofObject As Type
```

C++

```
public:  
static initonly Type^ s_typeofObject
```

F#

```
static val s_typeofObject: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofObjectActivator Field

Cache of typeof(ObjectActivator)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofObjectActivator
```

VB

```
Public Shared ReadOnly s_typeofObjectActivator As Type
```

C++

```
public:  
static initonly Type^ s_typeofObjectActivator
```

F#

```
static val s_typeofObjectActivator: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofObjectArray Field

Cache of typeof([Object\[\]](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofObjectArray
```

VB

```
Public Shared ReadOnly s_typeofObjectArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofObjectArray
```

F#

```
static val s_typeofObjectArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofOid Field

Cache of typeof(Oid)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofOid
```

VB

```
Public Shared ReadOnly s_typeofOid As Type
```

C++

```
public:  
static initonly Type^ s_typeofOid
```

F#

```
static val s_typeofOid: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofOidShort Field

Cache of typeof(OidShort)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofOidShort
```

VB

```
Public Shared ReadOnly s_typeofOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofOidShort
```

F#

```
static val s_typeofOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofOnePerDatabase Field

Cache of typeof([OnePerDatabase](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofOnePerDatabase
```

VB

```
Public Shared ReadOnly s_typeofOnePerDatabase As Type
```

C++

```
public:  
static initonly Type^ s_typeofOnePerDatabase
```

F#

```
static val s_typeofOnePerDatabase: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeOfOptimizedPersistable Field

Cache of typeof([OptimizedPersistable](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeOfOptimizedPersistable
```

VB

```
Public Shared ReadOnly s_typeOfOptimizedPersistable As Type
```

C++

```
public:  
static initonly Type^ s_typeOfOptimizedPersistable
```

F#

```
static val s_typeOfOptimizedPersistable: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofSByte Field

Cache of typeof([SByte](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofSByte
```

VB

```
Public Shared ReadOnly s_typeofSByte As Type
```

C++

```
public:  
static initonly Type^ s_typeofSByte
```

F#

```
static val s_typeofSByte: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofSchema Field

Cache of typeof(Schema)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofSchema
```

VB

```
Public Shared ReadOnly s_typeofSchema As Type
```

C++

```
public:  
static initonly Type^ s_typeofSchema
```

F#

```
static val s_typeofSchema: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofSessionBase Field

Cache of typeof(SessionBase)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofSessionBase
```

VB

```
Public Shared ReadOnly s_typeofSessionBase As Type
```

C++

```
public:  
static initonly Type^ s_typeofSessionBase
```

F#

```
static val s_typeofSessionBase: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofSetHandler Field

Cache of typeof(Action{object, object});

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofSetHandler
```

VB

```
Public Shared ReadOnly s_typeofSetHandler As Type
```

C++

```
public:  
static initonly Type^ s_typeofSetHandler
```

F#

```
static val s_typeofSetHandler: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofSingle Field

Cache of typeof([Single](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofSingle
```

VB

```
Public Shared ReadOnly s_typeofSingle As Type
```

C++

```
public:  
static initonly Type^ s_typeofSingle
```

F#

```
static val s_typeofSingle: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofString Field

Cache of typeof([String](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofString
```

VB

```
Public Shared ReadOnly s_typeofString As Type
```

C++

```
public:  
static initonly Type^ s_typeofString
```

F#

```
static val s_typeofString: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofStringArray Field

Cache of typeof([String\[\]](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofStringArray
```

VB

```
Public Shared ReadOnly s_typeofStringArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofStringArray
```

F#

```
static val s_typeofStringArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofTimeSpan Field

Cache of typeof([TimeSpan](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofTimeSpan
```

VB

```
Public Shared ReadOnly s_typeofTimeSpan As Type
```

C++

```
public:  
static initonly Type^ s_typeofTimeSpan
```

F#

```
static val s_typeofTimeSpan: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofType Field

Cache of typeof([Type](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofType
```

VB

```
Public Shared ReadOnly s_typeofType As Type
```

C++

```
public:  
static initonly Type^ s_typeofType
```

F#

```
static val s_typeofType: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofTypeArray Field

Cache of typeof([Type\[\]](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofTypeArray
```

VB

```
Public Shared ReadOnly s_typeofTypeArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofTypeArray
```

F#

```
static val s_typeofTypeArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofTypeVersionArray Field

Cache of typeof(TypeVersion[])

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofTypeVersionArray
```

VB

```
Public Shared ReadOnly s_typeofTypeVersionArray As Type
```

C++

```
public:  
static initonly Type^ s_typeofTypeVersionArray
```

F#

```
static val s_typeofTypeVersionArray: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofUInt16 Field

Cache of typeof([UInt16](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofUInt16
```

VB

```
Public Shared ReadOnly s_typeofUInt16 As Type
```

C++

```
public:  
static initonly Type^ s_typeofUInt16
```

F#

```
static val s_typeofUInt16: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofUInt32 Field

Cache of typeof([UInt32](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofUInt32
```

VB

```
Public Shared ReadOnly s_typeofUInt32 As Type
```

C++

```
public:  
static initonly Type^ s_typeofUInt32
```

F#

```
static val s_typeofUInt32: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofUInt64 Field

Cache of typeof([UInt64](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofUInt64
```

VB

```
Public Shared ReadOnly s_typeofUInt64 As Type
```

C++

```
public:  
static initonly Type^ s_typeofUInt64
```

F#

```
static val s_typeofUInt64: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofUniqueConstraint Field

Cache of typeof(UniqueConstraint)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofUniqueConstraint
```

VB

```
Public Shared ReadOnly s_typeofUniqueConstraint As Type
```

C++

```
public:  
static initonly Type^ s_typeofUniqueConstraint
```

F#

```
static val s_typeofUniqueConstraint: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofUseOidShort Field

Cache of typeof(UseOidShort)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofUseOidShort
```

VB

```
Public Shared ReadOnly s_typeofUseOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofUseOidShort
```

F#

```
static val s_typeofUseOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofValueType Field

Cache of typeof([ValueType](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofValueType
```

VB

```
Public Shared ReadOnly s_typeofValueType As Type
```

C++

```
public:  
static initonly Type^ s_typeofValueType
```

F#

```
static val s_typeofValueType: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofVelocityDbComparer Field

Cache of typeof([VelocityDbComparer\(Key\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofVelocityDbComparer
```

VB

```
Public Shared ReadOnly s_typeofVelocityDbComparer As Type
```

C++

```
public:  
static initonly Type^ s_typeofVelocityDbComparer
```

F#

```
static val s_typeofVelocityDbComparer: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofVelocityDbList Field

Cache of typeof([VelocityDbList\(T\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofVelocityDbList
```

VB

```
Public Shared ReadOnly s_typeofVelocityDbList As Type
```

C++

```
public:  
static initonly Type^ s_typeofVelocityDbList
```

F#

```
static val s_typeofVelocityDbList: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofVelocityDbListOidShort Field

Cache of typeof([VelocityDbListOidShort\(T\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofVelocityDbListOidShort
```

VB

```
Public Shared ReadOnly s_typeofVelocityDbListOidShort As Type
```

C++

```
public:  
static initonly Type^ s_typeofVelocityDbListOidShort
```

F#

```
static val s_typeofVelocityDbListOidShort: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofWeakIOptimizedPersistableReference Field

Cache of typeof(WeakIOptimizedPersistableReference)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofWeakIOptimizedPersistableReference
```

VB

```
Public Shared ReadOnly s_typeofWeakIOptimizedPersistableReference As Type
```

C++

```
public:  
static initonly Type^ s_typeofWeakIOptimizedPersistableReference
```

F#

```
static val s_typeofWeakIOptimizedPersistableReference: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofWeakIOptimizedPersistableReferenceBase Field

Cache of typeof(WeakIOptimizedPersistableReferenceBase)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofWeakIOptimizedPersistableReferenceBase
```

VB

```
Public Shared ReadOnly s_typeofWeakIOptimizedPersistableReferenceBase As Type
```

C++

```
public:  
static initonly Type^ s_typeofWeakIOptimizedPersistableReferenceBase
```

F#

```
static val s_typeofWeakIOptimizedPersistableReferenceBase: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofWeakReferenceList Field

Cache of typeof([WeakReferenceList\(T\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofWeakReferenceList
```

VB

```
Public Shared ReadOnly s_typeofWeakReferenceList As Type
```

C++

```
public:  
static initonly Type^ s_typeofWeakReferenceList
```

F#

```
static val s_typeofWeakReferenceList: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

CommonTypes.s_typeofWeakShortReferenceList Field

Cache of typeof([WeakShortReferenceList\(T\)](#))

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Type s_typeofWeakShortReferenceList
```

VB

```
Public Shared ReadOnly s_typeofWeakShortReferenceList As Type
```

C++

```
public:  
static initonly Type^ s_typeofWeakShortReferenceList
```

F#

```
static val s_typeofWeakShortReferenceList: Type
```

Field Value

Type: [Type](#)

See Also

[CommonTypes Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember Class

Info about a data Field of a persistent object

[Inheritance Hierarchy](#)

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.DataMember

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class DataMember : OptimizedPersistable, IEquatable<DataMember>
```

VB

```
<SerializableAttribute>
Public Class DataMember
    Inherits OptimizedPersistable
    Implements IEquatable(Of DataMember)
```

C++


```
[SerializableAttribute]
public ref class DataMember : public OptimizedPersistable,
    IEquatable<DataMember^>
```









F#

```
[<SerializableAttribute>]
type DataMember =
    class
        inherit OptimizedPersistable
        interface IEquatable<DataMember>
    end
```




















The **DataMember** type exposes the following members.

Properties




| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
|  Field | The managed field |
|  FieldName | Name of managed field |
|  FieldType | Declared Type of this field |
|  GetTypeCode | Get the type code for a field |
|  HasId | Does this field have an Oid? |
|  IsGuid | true if field is of type Guid |
|  IsInterface | true if field is if type interface |
|  WeakIOptimizedPersistableReference | Is this field a VelocityDb.WeakIOptimizedPersistableReference |



Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DecodeToArray  | Used by code generator for reading objects. |
|  DecodeToString  | Used by code generator. |
|  Equals | Determines whether two DataMember instances are equal. |
|  GetField | Get FieldInfo for the fields VelocityDB cares about |
|  GetMemberValue | Gets the value of a member within a specified object |
|  GetTypeFromAnyAssemblyVersion  | Load type using GetType(String) , and if fails, attempt to load same type from an assembly by assembly name, without specifying assembly version or any other part of the signature |
|  InitializeAfterRead | Sets up some transient fields (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase) .) |
|  isNull  | Determines if given bytes represents null |
|  isShortOidNull  | Determines if given bytes represent a null short reference |
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |
|  SetMemberValue | Sets the value of a field within an object |
|  SetMemberValueWithPossibleConvert | Internal Use and in VelocityDBExtensions |
|  StringToType  | Used by code generator. |

VelocityDB Class Library

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|  ToString | Override to provide more internal info in browser (Overrides OptimizedPersistable.ToString().) |
|  TypeToString  | Converts a type to a string the way VelocityDB represents it internally |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |










See Also

[VelocityDb.TypeInfo Namespace](#)

DataMember.DataMember Properties

The [DataMember](#) type exposes the following members.

Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  Field | The managed field |
|  FieldName | Name of managed field |
|  FieldType | Declared Type of this field |
|  GetTypeCode | Get the type code for a field |
|  HasId | Does this field have an Oid? |
|  IsGuid | true if field is of type Guid |
|  IsInterface | true if field is if type interface |
|  WeakIOptimizedPersistableReference | Is this field a VelocityDb.WeakIOptimizedPersistableReference |

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.Field Property

The managed field

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FieldInfo Field { get; }
```

VB

```
Public ReadOnly Property Field As FieldInfo  
    Get
```

C++

```
public:  
property FieldInfo^ Field {  
    FieldInfo^ get ();  
}
```

F#

```
member Field : FieldInfo with get
```

Property Value

Type: [FieldInfo](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.FieldName Property

Name of managed field

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string FieldName { get; }
```

VB

```
Public ReadOnly Property FieldName As String  
    Get
```

C++

```
public:  
property String^ FieldName {  
    String^ get ();  
}
```

F#

```
member FieldName : string with get
```

Property Value

Type: [String](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.FieldType Property

Declared [Type](#) of this field

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type FieldType { get; }
```

VB

```
Public ReadOnly Property FieldType As Type  
    Get
```

C++

```
public:  
property Type^ FieldType {  
    Type^ get ();  
}
```

F#

```
member FieldType : Type with get
```

Property Value

Type: [Type](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.GetTypeCode Property

Get the type code for a field

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeCode GetTypeCode { get; }
```

VB

```
Public ReadOnly Property GetTypeCode As TypeCode  
    Get
```

C++

```
public:  
property TypeCode GetTypeCode {  
    TypeCode get ();  
}
```

F#

```
member GetTypeCode : TypeCode with get
```

Property Value

Type: [TypeCode](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.HasId Property

Does this field have an Oid?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool HasId { get; }
```

VB

```
Public ReadOnly Property HasId As Boolean  
    Get
```

C++

```
public:  
property bool HasId {  
    bool get ();  
}
```

F#

```
member HasId : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.IsGuid Property

true

if field is of type [Guid](#)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsGuid { get; }
```

VB

```
Public ReadOnly Property IsGuid As Boolean  
    Get
```

C++

```
public:  
property bool IsGuid {  
    bool get ();  
}
```

F#

```
member IsGuid : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.IsInterface Property

true

if field is if type interface

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsInterface { get; }
```

VB

```
Public ReadOnly Property IsInterface As Boolean  
    Get
```

C++

```
public:  
property bool IsInterface {  
    bool get ();  
}
```

F#

```
member IsInterface : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.WeakIOptimizedPersistableReference Property

Is this field a VelocityDb.WeakIOptimizedPersistableReference

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool WeakIOptimizedPersistableReference { get; }
```

VB

```
Public ReadOnly Property WeakIOptimizedPersistableReference As Boolean  
    Get
```

C++

```
public:  
property bool WeakIOptimizedPersistableReference {  
    bool get ();  
}
```

F#

```
member WeakIOptimizedPersistableReference : bool with get
```

Property Value

Type: [Boolean](#)

See Also














[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)


DataMember.DataMember Methods

The [DataMember](#) type exposes the following members.


Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DecodeToArray | Used by code generator for reading objects. |
|  DecodeToString | Used by code generator. |
|  Equals | Determines whether two DataMember instances are equal. |
|  GetField | Get FieldInfo for the fields VelocityDB cares about |
|  GetMemberValue | Gets the value of a member within a specified object |
|  GetTypeFromAnyAssemblyVersion | Load type using GetType(String) , and if fails, attempt to load same type from an assembly by assembly name, without specifying assembly version or any other part of the signature |
|  InitializeAfterRead | Sets up some transient fields (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase) .) |
|  isNull | Determines if given bytes represents null |
|  isShortOidNull | Determines if given bytes represent a null short reference |
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |
|  SetMemberValue | Sets the value of a field within an object |
|  SetMemberValueWithPossibleConvert | Internal Use and in VelocityDBExtensions |
|  StringToType | Used by code generator. |
|  ToString | Override to provide more internal info in browser (Overrides OptimizedPersistable.ToString() .) |
|  TypeToString | Converts a type to a string the way VelocityDB represents it internally |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.DecodeToArray Method

Used by code generator for reading objects.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Object DecodeToArray(
    byte[] memberBytes,
    Type type,
    ref int offset,
    IOptimizedPersistable o,
    Page page,
    bool useOidShort,
    bool openRefs,
    Schema schema,
    SessionBase session,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly,
    bool embed
)
```

VB

```
Public Shared Function DecodeToArray (
    memberBytes As Byte(),
    type As Type,
    ByRef offset As Integer,
    o As IOptimizedPersistable,
    page As Page,
    useOidShort As Boolean,
    openRefs As Boolean,
    schema As Schema,
    session As SessionBase,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean,
    embed As Boolean
) As Object
```

C++

```
public:
static Object^ DecodeToArray(
    array<unsigned char>^ memberBytes,
    Type^ type,
    int% offset,
    IOptimizedPersistable^ o,
    Page^ page,
```

```
bool useOidShort,  
bool openRefs,  
Schema^ schema,  
SessionBase^ session,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly,  
bool embed  
)
```

F#

```
static member DecodeToArray :  
    memberBytes : byte[] *  
    type : Type *  
    offset : int byref *  
    o : IOptimizedPersistable *  
    page : Page *  
    useOidShort : bool *  
    openRefs : bool *  
    schema : Schema *  
    session : SessionBase *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool *  
    embed : bool -> Object
```

Parameters

memberBytes

Type: [System.Byte\[\]](#)

Bytes of the object to be read

type

Type: [System.Type](#)

The [Type](#) of the array.

offset

Type: [System.Int32](#)

Current offset into array of bytes

o

Type: [VelocityDb.IOptimizedPersistable](#)

Object containing the array

page

Type: [VelocityDb.Page](#)

The [Page](#) that the array belongs to.

useOidShort

VelocityDB Class Library

Type: [System.Boolean](#)

Use short id (page-slot) for object references

openRefs

Type: [System.Boolean](#)

Open referenced objects?

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active [Schema](#)

session

Type: [VelocityDb.Session.SessionBase](#)

The active session to use

toLoadMembers

Type: [System.Collections.Generic.List\(IOptimizedPersistable\)](#)

Keeps track of fields to load

graphDepth

Type: [System.Int32](#)

Current graph depth

graphDepthToLoad

Type: [System.Int32](#)

How many levels to load

primitivesOnly

Type: [System.Boolean](#)

Load only primitive fields>

embed

Type: [System.Boolean](#)

[Missing <param name="embed"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.DecodeToArray(System.Byte[],System.Type,System.Int32@,VelocityDb.IOptimizedPersistable,VelocityDb.Page,System.Boolean,System.Boolean,VelocityDb.TypeInfo.Schema,VelocityDb.Session.SessionBase,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean,System.Boolean)"]

Return Value

Type: [Object](#)

The decoded array

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.DecodeToString Method

Used by code generator.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string DecodeToString(  
    byte[] member,  
    ref int offset,  
    int numberOfBytes  
)
```

VB

```
Public Shared Function DecodeToString (  
    member As Byte(),  
    ByRef offset As Integer,  
    numberOfBytes As Integer  
) As String
```

C++

```
public:  
static String^ DecodeToString(  
    array<unsigned char>^ member,  
    int% offset,  
    int numberOfBytes  
)
```

F#

```
static member DecodeToString :  
    member : byte[] *  
    offset : int byref *  
    numberOfBytes : int -> string
```

Parameters

member

Type: [System.Byte\[\]](#)

Bytes for an object field

offset

Type: [System.Int32](#)

Offset into bytes

numberOfBytes

Type: [System.Int32](#)

Length of field

VelocityDB Class Library

Return Value

Type: [String](#)

A string representing the field value

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.Equals Method

Determines whether two [DataMember](#) instances are equal.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    DataMember other  
)
```

VB

```
Public Function Equals (  
    other As DataMember  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    DataMember^ other  
) sealed
```

F#

```
abstract Equals :  
    other : DataMember -> bool  
override Equals :  
    other : DataMember -> bool
```

Parameters

other

Type: [VelocityDb.TypeInfo.DataMember](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

true if the specified [DataMember](#) is equal to the current [DataMember](#); otherwise, false.

Implements

[IEquatable\(T\).Equals\(T\)](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.GetField Method

Get [FieldInfo](#) for the fields VelocityDB cares about

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FieldInfo GetField(  
    Type t  
)
```

VB

```
Public Function GetField (  
    t As Type  
) As FieldInfo
```

C++

```
public:  
FieldInfo^ GetField(  
    Type^ t  
)
```

F#

```
member GetField :  
    t : Type -> FieldInfo
```

Parameters

t

Type: [System.Type](#)

The type we are requesting it for

Return Value

Type: [FieldInfo](#)

[FieldInfo](#) for the fields VelocityDB cares about

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.GetMemberValue Method

Gets the value of a member within a specified object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetMemberValue (
    Object obj
)
```

VB

```
Public Function GetMemberValue (
    obj As Object
) As Object
```

C++

```
public:
Object^ GetMemberValue (
    Object^ obj
)
```

F#

```
member GetMemberValue :
    obj : Object -> Object
```

Parameters

obj

Type: [System.Object](#)

The object containing the field

Return Value

Type: [Object](#)

The value of the field

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.GetTypeFromAnyAssemblyVersion Method

Load type using [GetType\(String\)](#), and if fails, attempt to load same type from an assembly by assembly name, without specifying assembly version or any other part of the signature

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Type GetTypeFromAnyAssemblyVersion(  
    string typeName  
)
```

VB

```
Public Shared Function GetTypeFromAnyAssemblyVersion (  
    typeName As String  
) As Type
```

C++

```
public:  
static Type^ GetTypeFromAnyAssemblyVersion(  
    String^ typeName  
)
```

F#

```
static member GetTypeFromAnyAssemblyVersion :  
    typeName : string -> Type
```

Parameters

typeName

Type: [System.String](#)

The assembly-qualified name of the type to get. See [System.Type.AssemblyQualifiedName](#). If the type is in the currently executing assembly or in Mscorlib.dll, it is sufficient to supply the type name qualified by its namespace.

Return Value

Type: [Type](#)

A [Type](#) as decoded from [String](#).

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.InitializeAfterRead Method

Sets up some transient fields

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object.

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.isNull Method

Determines if given bytes represents null

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool isNull(  
    byte[] memberBytes,  
    int offset  
)
```

VB

```
Public Shared Function isNull (  
    memberBytes As Byte(),  
    offset As Integer  
) As Boolean
```

C++

```
public:  
static bool isNull(  
    array<unsigned char>^ memberBytes,  
    int offset  
)
```

F#

```
static member isNull :  
    memberBytes : byte[] *  
    offset : int -> bool
```

Parameters

memberBytes

Type: [System.Byte\[\]](#)

bytes to check

offset

Type: [System.Int32](#)

offset into bytes

Return Value

Type: [Boolean](#)

true if bytes represent null; otherwise false

See Also

[DataMember Class](#)

DataMember.isShortOidNull Method

Determines if given bytes represent a null short reference

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool isShortOidNull(  
    byte[] memberBytes,  
    int offset  
)
```

VB

```
Public Shared Function isShortOidNull (  
    memberBytes As Byte(),  
    offset As Integer  
) As Boolean
```

C++

```
public:  
static bool isShortOidNull(  
    array<unsigned char>^ memberBytes,  
    int offset  
)
```

F#

```
static member isShortOidNull :  
    memberBytes : byte[] *  
    offset : int -> bool
```

Parameters

memberBytes

Type: [System.Byte\[\]](#)

bytes to check

offset

Type: [System.Int32](#)

offset into bytes

Return Value

Type: [Boolean](#)

true if bytes represent null; otherwise false

See Also

[DataMember Class](#)

VelocityDB Class Library

[VelocityDb.TypeInfo Namespace](#)

DataMember.ReadMe Method

Used by code generator

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
```

```
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
    ) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Sche

`ma,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]`

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List{IOptimizedPersistable}](#)

[Missing <param name="toLoadMembers"/> documentation for

"M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System

m.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.TypeInfo.DataMember.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List\(IOptimizedPersistable\), Int32, Int32, Boolean\)](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.SetMemberValue Method

Sets the value of a field within an object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetMemberValue (  
    Object obj,  
    Object value  
)
```

VB

```
Public Sub SetMemberValue (  
    obj As Object,  
    value As Object  
)
```

C++

```
public:  
void SetMemberValue (  
    Object^ obj,  
    Object^ value  
)
```

F#

```
member SetMemberValue :  
    obj : Object *  
    value : Object -> unit
```

Parameters

obj

Type: [System.Object](#)

Object to set field value for

value

Type: [System.Object](#)

The field value

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.SetMemberValueWithPossibleConvert Method

Internal Use and in VelocityDBExtensions

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public void SetMemberValueWithPossibleConvert (  
    Object obj,  
    Object value  
)
```

```
VB  
Public Sub SetMemberValueWithPossibleConvert (  
    obj As Object,  
    value As Object  
)
```

```
C++  
public:  
void SetMemberValueWithPossibleConvert (  
    Object^ obj,  
    Object^ value  
)
```

```
F#  
member SetMemberValueWithPossibleConvert :  
    obj : Object *  
    value : Object -> unit
```

Parameters

obj

Type: [System.Object](#)

Object for which to set a value

value

Type: [System.Object](#)

Value to set to

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.StringToType Method

Used by code generator.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Type StringToType(  
    string typeAsString,  
    SessionBase session,  
    out bool typeUpdated  
)
```

VB

```
Public Shared Function StringToType (  
    typeAsString As String,  
    session As SessionBase,  
    <OutAttribute> ByRef typeUpdated As Boolean  
) As Type
```

C++

```
public:  
static Type^ StringToType(  
    String^ typeAsString,  
    SessionBase^ session,  
    [OutAttribute] bool% typeUpdated  
)
```

F#

```
static member StringToType :  
    typeAsString : string *  
    session : SessionBase *  
    typeUpdated : bool byref -> Type
```

Parameters

typeAsString

Type: [System.String](#)

A [Type](#) encoded as a [String](#).

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

typeUpdated

Type: [System.Boolean](#)

Was a [Type](#) change detected?

VelocityDB Class Library

Return Value

Type: [Type](#)

Decoded [Type](#).

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.ToString Method

Override to provide more internal info in browser

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

"DataMember " + fieldName + " " + [Oid.AsString\(id\)](#)

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

DataMember.TypeToString Method

Converts a type to a string the way VelocityDB represents it internally

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string TypeToString(  
    Type type  
)
```

VB

```
Public Shared Function TypeToString (  
    type As Type  
) As String
```

C++

```
public:  
static String^ TypeToString(  
    Type^ type  
)
```

F#

```
static member TypeToString :  
    type : Type -> string
```

Parameters

type

Type: [System.Type](#)

[Missing <param name="type"/> documentation for "M:VelocityDb.TypeInfo.DataMember.TypeToString(System.Type)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDb.TypeInfo.DataMember.TypeToString(System.Type)"]

See Also

[DataMember Class](#)

[VelocityDb.TypeInfo Namespace](#)

FieldAccessor Class

A field attribute that controls how many objects that will be placed on the same page as the object of the field.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.TypeInfo.FieldAccessor

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class FieldAccessor : Attribute
```

VB

```
Public NotInheritable Class FieldAccessor
    Inherits Attribute
```

C++


```
public ref class FieldAccessor sealed : public Attribute
```

F#


```
[<SealedAttribute>]
type FieldAccessor =
    class
        inherit Attribute
    end
```

The **FieldAccessor** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | FieldAccessor | Use in class definition preceding a property declaration, i.e. [FieldAccessor("myProtectedField")] where myProtectedField is the field name of the field returned by the property. |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------|
|  | FieldName | Gets the field name of the field being accessed by the property |

VelocityDB Class Library

See Also

[VelocityDb.TypeInfo Namespace](#)

FieldAccessor Constructor

Use in class definition preceding a property declaration, i.e. [FieldAccessor("myProtectedField")] where myProtectedField is the field name of the field returned by the property.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public FieldAccessor(  
    string fieldName  
)
```

VB

```
Public Sub New (  
    fieldName As String  
)
```

C++

```
public:  
FieldAccessor(  
    String^ fieldName  
)
```

F#

```
new :  
    fieldName : string -> FieldAccessor
```

Parameters

fieldName

Type: [System.String](#)

Name of field.

See Also


[FieldAccessor Class](#)

[VelocityDb.TypeInfo Namespace](#)

FieldAccessor.FieldAccessor Properties

The [FieldAccessor](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------|
|  | FieldName | Gets the field name of the field being accessed by the property |

See Also

[FieldAccessor Class](#)

[VelocityDb.TypeInfo Namespace](#)

FieldAccessor.FieldName Property

Gets the field name of the field being accessed by the property

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string FieldName { get; }
```

VB

```
Public ReadOnly Property FieldName As String  
    Get
```

C++

```
public:  
property String^ FieldName {  
    String^ get ();  
}
```

F#

```
member FieldName : string with get
```

Property Value

Type: [String](#)

The name of the field.

See Also

[FieldAccessor Class](#)

[VelocityDb.TypeInfo Namespace](#)

ObjectsPerPage Class

A field attribute that controls how many objects that will be placed on the same page as the object of the field.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.TypeInfo.ObjectsPerPage

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class ObjectsPerPage : Attribute
```

VB

```
Public NotInheritable Class ObjectsPerPage  
    Inherits Attribute
```

C++


```
public ref class ObjectsPerPage sealed : public Attribute
```

F#


```
[<SealedAttribute>]  
type ObjectsPerPage =  
    class  
        inherit Attribute  
    end
```

The **ObjectsPerPage** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------|
|  | ObjectsPerPage | Use in class definition preceding a field declaration, i.e. [ObjectsPerPage(1)] |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------|
|  | LimitPerPage | Gets the requested maximum number of objects on the page |

See Also

[VelocityDb.TypeInfo Namespace](#)

ObjectsPerPage Constructor

Use in class definition preceding a field declaration, i.e. [ObjectsPerPage(1)]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ObjectsPerPage(  
    ushort limitPerPage  
)
```

VB

```
Public Sub New (  
    limitPerPage As UShort  
)
```

C++

```
public:  
ObjectsPerPage(  
    unsigned short limitPerPage  
)
```

F#

```
new :  
    limitPerPage : uint16 -> ObjectsPerPage
```

Parameters

limitPerPage

Type: [System.UInt16](#)

The requested maximum number of objects on the page.

See Also


[ObjectsPerPage Class](#)

[VelocityDb.TypeInfo Namespace](#)

ObjectsPerPage.ObjectsPerPage Properties

The [ObjectsPerPage](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------|
|  | LimitPerPage | Gets the requested maximum number of objects on the page |

See Also

[ObjectsPerPage Class](#)

[VelocityDb.TypeInfo Namespace](#)

ObjectsPerPage.LimitPerPage Property

Gets the requested maximum number of objects on the page

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ushort LimitPerPage { get; }
```

VB

```
Public ReadOnly Property LimitPerPage As UShort  
    Get
```

C++

```
public:  
property unsigned short LimitPerPage {  
    unsigned short get ();  
}
```

F#

```
member LimitPerPage : uint16 with get
```

Property Value

Type: [UInt16](#)

The [UInt16](#) number of objects.

See Also

[ObjectsPerPage Class](#)

[VelocityDb.TypeInfo Namespace](#)

Reference Class

[Missing <summary> documentation for "T:VelocityDb.TypeInfo.Reference"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.Reference

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Reference : OptimizedPersistable, IComparable<Reference>
```

VB

```
Public Class Reference
    Inherits OptimizedPersistable
    Implements IComparable(Of Reference)
```

C++


```
public ref class Reference : public OptimizedPersistable,
    IComparable<Reference^>
```

F#



```
type Reference =
    class
        inherit OptimizedPersistable
        interface IComparable<Reference>
    end
```

The **Reference** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------|
|  | Reference | Initializes a new instance of the Reference class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------|
|  | RefFieldName | Field name in To object that contains reference |
|  | To | Persistent object being referenced |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.TypeInfo Namespace](#)

Reference Constructor

Initializes a new instance of the [Reference](#) class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Reference (
    IOptimizedPersistable aRef,
    string refFieldName
)
```

VB

```
Public Sub New (
    aRef As IOptimizedPersistable,
    refFieldName As String
)
```

C++

```
public:
Reference (
    IOptimizedPersistable^ aRef,
    String^ refFieldName
)
```

F#

```
new :
    aRef : IOptimizedPersistable *
    refFieldName : string -> Reference
```

Parameters

aRef

Type: [VelocityDb.IOptimizedPersistable](#)

[Missing <param name="aRef"/> documentation for

"M:VelocityDb.TypeInfo.Reference.#ctor(VelocityDb.IOptimizedPersistable,System.String)"]

refFieldName

Type: [System.String](#)

[Missing <param name="refFieldName"/> documentation for

"M:VelocityDb.TypeInfo.Reference.#ctor(VelocityDb.IOptimizedPersistable,System.String)"]

See Also



[Reference Class](#)

[VelocityDb.TypeInfo Namespace](#)

Reference.Reference Properties

The [Reference](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------------------------------------------|
|  | RefFieldName | Field name in To object that contains reference |
|  | To | Persistent object being referenced |

See Also

[Reference Class](#)

[VelocityDb.TypeInfo Namespace](#)

Reference.RefFieldName Property

Field name in To object that contains reference

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string RefFieldName { get; }
```

VB

```
Public ReadOnly Property RefFieldName As String  
    Get
```

C++

```
public:  
property String^ RefFieldName {  
    String^ get ();  
}
```

F#

```
member RefFieldName : string with get
```

Property Value

Type: [String](#)

See Also

[Reference Class](#)

[VelocityDb.TypeInfo Namespace](#)

Reference.To Property

Persistent object being referenced

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IOptimizedPersistable To { get; }
```

VB

```
Public ReadOnly Property To As IOptimizedPersistable  
    Get
```

C++

```
public:  
property IOptimizedPersistable^ To {  
    IOptimizedPersistable^ get ();  
}
```

F#

```
member To : IOptimizedPersistable with get
```

Property Value

Type: [IOptimizedPersistable](#)

See Also



[Reference Class](#)

[VelocityDb.TypeInfo Namespace](#)

Reference.Reference Methods

The [Reference](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[Reference Class](#)

[VelocityDb.TypeInfo Namespace](#)

Relation Class

The relation classes exist as an aid to maintain referential integrity, that is no dangling references (references to deleted objects). NOTE: These classes may change as they are still in a prototype phase. Let us know how we can improve them? Any ideas are appreciated!

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.Relation

[VelocityDb.TypeInfo.RelationManyToMany\(From, To\)](#)

[VelocityDb.TypeInfo.RelationOneToMany\(From, To\)](#)

[VelocityDb.TypeInfo.RelationOneToOne\(From, To\)](#)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public abstract class Relation : OptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public MustInherit Class Relation  
    Inherits OptimizedPersistable
```

C++

```
[SerializableAttribute]  
public ref class Relation abstract : public OptimizedPersistable
```

F#

```
[<AbstractClassAttribute>]  
[<SerializableAttribute>]  
type Relation =  
    class  
        inherit OptimizedPersistable  
    end
```

See Also

[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(From, To) Class

Use for many to many relations

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.TypeInfo.Relation](#)

VelocityDb.TypeInfo.RelationManyToMany(From, To)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class RelationManyToMany<From, To> : Relation  
where From : IOptimizedPersistable  
where To : IOptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class RelationManyToMany(Of From As IOptimizedPersistable, To As  
IOptimizedPersistable)  
    Inherits Relation
```

C++

```
[SerializableAttribute]  
generic<typename From, typename To>  
where From : IOptimizedPersistable  
where To : IOptimizedPersistable  
public ref class RelationManyToMany : public Relation
```

F#

```
[<SerializableAttribute>]  
type RelationManyToMany<'From, 'To when 'From : IOptimizedPersistable when  
'To : IOptimizedPersistable> =  
    class  
        inherit Relation  
    end
```

Type Parameters

From


Type of From objects

To



Type of To objects

The RelationManyToMany(From, To) type exposes the following members.


Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|--------------------------------|
|  | RelationManyToMany(From, To) | Use for many to many relations |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|--------------------------|
|  | RelatedFrom | Returns the From objects |
|  | RelatedTo | Returns the To objects |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(*From*, *To*) Constructor

Use for many to many relations

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public RelationManyToMany(  
    string fromFieldName,  
    string toFieldName,  
    SessionBase session,  
    BTreeSet<From> from = null,  
    BTreeSet<To> to = null  
)
```

VB

```
Public Sub New (  
    fromFieldName As String,  
    toFieldName As String,  
    session As SessionBase,  
    Optional from As BTreeSet(Of From) = Nothing,  
    Optional to As BTreeSet(Of To) = Nothing  
)
```

C++

```
public:  
RelationManyToMany(  
    String^ fromFieldName,  
    String^ toFieldName,  
    SessionBase^ session,  
    BTreeSet<From>^ from = nullptr,  
    BTreeSet<To>^ to = nullptr  
)
```

F#

```
new :  
    fromFieldName : string *  
    toFieldName : string *  
    session : SessionBase *  
    ?from : BTreeSet<'From> *  
    ?to : BTreeSet<'To>  
(* Defaults:  
    let_from = defaultArg from null  
    let_to = defaultArg to null  
)  
-> RelationManyToMany
```


VelocityDB Class Library

Parameters

fromFieldName

Type: [System.String](#)

Field name in from object that references this relation

toFieldName

Type: [System.String](#)

Field name in to object that references this relation

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

from (Optional)

Type: [VelocityDb.Collection.BTree.BTreeSet\(From\)](#)

Collection of From objects. If null, default [BTreeSet\(Key\)](#) is created.

to (Optional)

Type: [VelocityDb.Collection.BTree.BTreeSet\(To\)](#)

Collection of To objects. If null, default is created.

See Also



[RelationManyToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(*From*, *To*).RelationManyToMany(*From*, *To*) Properties

The [RelationManyToMany\(*From*, *To*\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|--------------------------|
|  | RelatedFrom | Returns the From objects |
|  | RelatedTo | Returns the To objects |

See Also

[RelationManyToMany\(*From*, *To*\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(From, To).RelatedFrom Property

Returns the From objects

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<From> RelatedFrom { get; }
```

VB

```
Public ReadOnly Property RelatedFrom As BTreeSet(Of From)  
    Get
```

C++

```
public:  
property BTreeSet<From>^ RelatedFrom {  
    BTreeSet<From>^ get ();  
}
```

F#

```
member RelatedFrom : BTreeSet<'From> with get
```

Property Value

Type: [BTreeSet\(From\)](#)

See Also

[RelationManyToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(From, To).RelatedTo Property

Returns the To objects

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<To> RelatedTo { get; }
```

VB

```
Public ReadOnly Property RelatedTo As BTreeSet(Of To)  
    Get
```

C++

```
public:  
property BTreeSet<To>^ RelatedTo {  
    BTreeSet<To>^ get ();  
}
```

F#

```
member RelatedTo : BTreeSet<'To> with get
```

Property Value

Type: [BTreeSet\(To\)](#)

See Also


[RelationManyToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)



RelationManyToMany(*From*, *To*).RelationManyToMany(*From*, *To*) Methods

The [RelationManyToMany\(*From*, *To*\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[RelationManyToMany\(*From*, *To*\)Class](#)
[VelocityDb.TypeInfo Namespace](#)

RelationManyToMany(*From*, *To*).Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[RelationManyToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To) Class

Use for one to many (and many to one) relations.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.TypeInfo.Relation](#)

VelocityDb.TypeInfo.RelationOneToMany(From, To)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class RelationOneToMany<From, To> : Relation
where From : IOptimizedPersistable
where To : IOptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class RelationOneToMany(Of From As IOptimizedPersistable, To As
IOptimizedPersistable)
    Inherits Relation
```

C++

```
[SerializableAttribute]
generic<typename From, typename To>
where From : IOptimizedPersistable
where To : IOptimizedPersistable
public ref class RelationOneToMany : public Relation
```

F#

```
[<SerializableAttribute>]
type RelationOneToMany<'From, 'To when 'From : IOptimizedPersistable when 'To
: IOptimizedPersistable> =
    class
        inherit Relation
    end
```

Type Parameters

From

Type

of From object


To

[Type](#)




of To object

The RelationOneToMany(From, To) type exposes the following members.




Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|----------------------------------------------------|
|  | RelationOneToMany(From, To) | Creates the relations between From and To objects. |



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------|
|  | FromFieldName | Name of field referencing this object from From object |
|  | RelatedFrom | Gets or sets the From object |
|  | RelatedTo | Returns the To objects |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddRelationTo | Adds a new relations to a To object. Added to "many" collection. |
|  | RemoveRelationTo | Removes relation to a To object. Removed from "many" collection. |
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(*From*, *To*) Constructor

Creates the relations between From and To objects.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public RelationOneToMany(  
    string fromFieldName,  
    SessionBase session,  
    From from,  
    BTreeSet<To> to = null  
)
```

VB

```
Public Sub New (  
    fromFieldName As String,  
    session As SessionBase,  
    from As From,  
    Optional to As BTreeSet(Of To) = Nothing  
)
```

C++

```
public:  
RelationOneToMany(  
    String^ fromFieldName,  
    SessionBase^ session,  
    From from,  
    BTreeSet<To>^ to = nullptr  
)
```

F#

```
new :  
    fromFieldName : string *  
    session : SessionBase *  
    from : 'From *  
    ?to : BTreeSet<'To>  
(* Defaults:  
    let _to = defaultArg to null  
)  
-> RelationOneToMany
```

Parameters

fromFieldName

Type: [System.String](#)

Field name in from object that references this relation

session

VelocityDB Class Library

Type: [VelocityDb.Session.SessionBase](#)

The active session

from

Type: *From*

From object

to (Optional)

Type: [VelocityDb.Collection.BTree.BTreeSet\(To\)](#)

Collection of To objects. If null, default [BTreeSet\(Key\)](#) is created.

See Also

[RelationOneToMany\(From, To\)Class](#)




[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(*From*, *To*).RelationOneToMany(*From*, *To*)

Properties

The [RelationOneToMany\(*From*, *To*\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------|
|  | FromFieldName | Name of field referencing this object from From object |
|  | RelatedFrom | Gets or sets the From object |
|  | RelatedTo | Returns the To objects |

See Also

[RelationOneToMany\(*From*, *To*\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).FromFieldName Property

Name of field referencing this object from From object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string FromFieldName { get; }
```

VB

```
Public ReadOnly Property FromFieldName As String  
    Get
```

C++

```
public:  
property String^ FromFieldName {  
    String^ get ();  
}
```

F#

```
member FromFieldName : string with get
```

Property Value

Type: [String](#)

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).RelatedFrom Property

Gets or sets the From object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public From RelatedFrom { get; set; }
```

VB

```
Public Property RelatedFrom As From  
    Get  
    Set
```

C++

```
public:  
property From RelatedFrom {  
    From get ();  
    void set (From value);  
}
```

F#

```
member RelatedFrom : 'From with get, set
```

Property Value

Type: *From*

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).RelatedTo Property

Returns the To objects

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<To> RelatedTo { get; }
```

VB

```
Public ReadOnly Property RelatedTo As BTreeSet(Of To)  
    Get
```

C++

```
public:  
property BTreeSet<To>^ RelatedTo {  
    BTreeSet<To>^ get ();  
}
```

F#

```
member RelatedTo : BTreeSet<'To> with get
```

Property Value

Type: [BTreeSet\(To\)](#)

See Also




[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)



RelationOneToMany(From, To).RelationOneToMany(From, To) Methods

The [RelationOneToMany\(From, To\)](#) generic type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddRelationTo | Adds a new relations to a To object. Added to "many" collection. |
|  RemoveRelationTo | Removes relation to a To object. Removed from "many" collection. |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).AddRelationTo Method

Adds a new relations to a To object. Added to "many" collection.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddRelationTo(  
    To to  
)
```

VB

```
Public Sub AddRelationTo (  
    to As To  
)
```

C++

```
public:  
void AddRelationTo(  
    To to  
)
```

F#

```
member AddRelationTo :  
    to : 'To -> unit
```

Parameters

to

Type: *To*

To object to be added

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).RemoveRelationTo Method

Removes relation to a To object. Removed from "many" collection.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool RemoveRelationTo(  
    To to  
)
```

VB

```
Public Function RemoveRelationTo (  
    to As To  
) As Boolean
```

C++

```
public:  
bool RemoveRelationTo(  
    To to  
)
```

F#

```
member RemoveRelationTo :  
    to : 'To -> bool
```

Parameters

to

Type: *To*

To object to be removed.

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDb.TypeInfo.RelationOneToMany`2.RemoveRelationTo(`1)"]

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToMany(From, To).Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[RelationOneToMany\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(*From*, *To*) Class

Use for one to one relations.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityDb.TypeInfo.Relation](#)

VelocityDb.TypeInfo.RelationOneToOne(*From*, *To*)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class RelationOneToOne<From, To> : Relation  
where From : IOptimizedPersistable  
where To : IOptimizedPersistable
```

VB

```
<SerializableAttribute>  
Public Class RelationOneToOne (Of From As IOptimizedPersistable, To As  
IOptimizedPersistable)  
    Inherits Relation
```

C++

```
[SerializableAttribute]  
generic<typename From, typename To>  
where From : IOptimizedPersistable  
where To : IOptimizedPersistable  
public ref class RelationOneToOne : public Relation
```

F#

```
[<SerializableAttribute>]  
type RelationOneToOne<'From, 'To when 'From : IOptimizedPersistable when 'To  
: IOptimizedPersistable> =  
    class  
        inherit Relation  
    end
```

Type Parameters

From


Type of From object

To





Type of To object

The RelationOneToOne(From, To) type exposes the following members.



Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------|
|  | RelationOneToOne(From, To) | Creates the relations between From and To objects |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------|
|  | FromFieldName | Name of field referencing this object from From object |
|  | RelatedFrom | Returns the From object |
|  | RelatedTo | Returns the To object |
|  | ToFieldName | Name of field referencing this object from To object |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(*From*, *To*) Constructor

Creates the relations between From and To objects

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public RelationOneToOne(  
    From from,  
    To to,  
    string fromFieldName,  
    string toFieldName,  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    from As From,  
    to As To,  
    fromFieldName As String,  
    toFieldName As String,  
    session As SessionBase  
)
```

C++

```
public:  
RelationOneToOne(  
    From from,  
    To to,  
    String^ fromFieldName,  
    String^ toFieldName,  
    SessionBase^ session  
)
```

F#

```
new :  
    from : 'From *  
    to : 'To *  
    fromFieldName : string *  
    toFieldName : string *  
    session : SessionBase -> RelationOneToOne
```

Parameters

from

Type: *From*

From object

to

VelocityDB Class Library

Type: *To*
to object

fromFieldName

Type: [System.String](#)

Field name in from object that references this relation

toFieldName

Type: [System.String](#)

Field name in to object that references this relation

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

See Also





[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(From, To).RelationOneToOne(From, To) Properties

The [RelationOneToOne\(From, To\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------|
|  | FromFieldName | Name of field referencing this object from From object |
|  | RelatedFrom | Returns the From object |
|  | RelatedTo | Returns the To object |
|  | ToFieldName | Name of field referencing this object from To object |

See Also

[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(From, To).FromFieldName Property

Name of field referencing this object from From object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string FromFieldName { get; }
```

VB

```
Public ReadOnly Property FromFieldName As String  
    Get
```

C++

```
public:  
property String^ FromFieldName {  
    String^ get ();  
}
```

F#

```
member FromFieldName : string with get
```

Property Value

Type: [String](#)

See Also

[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(From, To).RelatedFrom Property

Returns the From object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public From RelatedFrom { get; set; }
```

VB

```
Public Property RelatedFrom As From  
    Get  
    Set
```

C++

```
public:  
property From RelatedFrom {  
    From get ();  
    void set (From value);  
}
```

F#

```
member RelatedFrom : 'From with get, set
```

Property Value

Type: *From*

See Also

[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(From, To).RelatedTo Property

Returns the To object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public To RelatedTo { get; set; }
```

VB

```
Public Property RelatedTo As To  
    Get  
    Set
```

C++

```
public:  
property To RelatedTo {  
    To get ();  
    void set (To value);  
}
```

F#

```
member RelatedTo : 'To with get, set
```

Property Value

Type: *To*

See Also

[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(From, To).ToFieldName Property

Name of field referencing this object from To object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string ToFieldName { get; }
```

VB

```
Public ReadOnly Property ToFieldName As String  
    Get
```

C++

```
public:  
property String^ ToFieldName {  
    String^ get ();  
}
```

F#

```
member ToFieldName : string with get
```

Property Value

Type: [String](#)

See Also



[RelationOneToOne\(From, To\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

RelationOneToOne(*From*, *To*).RelationOneToOne(*From*, *To*) Methods

The [RelationOneToOne\(*From*, *To*\)](#) generic type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[RelationOneToOne\(*From*, *To*\)Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema Class

Holds information about persistently stored types. Used internally by VelocityDb.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.Schema

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class Schema : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class Schema
    Inherits OptimizedPersistable
```

C++




```
[SerializableAttribute]
public ref class Schema : public OptimizedPersistable
```




F#

```
[<SerializableAttribute>]
type Schema =
    class
        inherit OptimizedPersistable
    end
```








The **Schema** type exposes the following members.

Properties







| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | Cache | Do cache CacheEnum.Yes (Overrides OptimizedPersistable.Cache.) |
|  | LookupByType | For use in VelocityDBExtension DLL |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  TypesByName | All persisted types ordered by Type name |
|  WeakReferencedTypes  | Info about what Type another Type is referencing indirectly via object Id reference or other non explicit/weak reference. If your class persists objects with implicit/weak references to other persisted objects then register this fact by adding to this Dictionary(TKey, TValue) . Information provided by this Dictionary(TKey, TValue) is primarily used by Database Manager Schema Connectivity Window. |


Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetTypeVersion | Opens a TypeVersion object |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase) .) |
|  IsExpandedInternalType | Includes additional build in types |
|  IsInternalType(UInt16) | Determine if a given slot is within the range of slots containing the build in/pre registered types |
|  IsInternalType(VelocityDbType) | Determine if a given VelocityDbType is a build in/pre registered type |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |
|  RegisterClass | Register a type in the persistent schema |


Fields

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
|  s_bootupTypeCount  | VelocityDB internal schema requires this many slots on schema page |
|  s_bootupTypeCountExpanded  | Additional internal schema added September 9, 2017 |
|  SchemaDB  | |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|







See Also

[VelocityDb.TypeInfo Namespace](#)

Schema.Schema Properties

The [Schema](#) type exposes the following members.

Properties

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | Cache | Do cache CacheEnum.Yes (Overrides OptimizedPersistable.Cache.) |
|  | LookupByType | For use in VelocityDBExtension DLL |
|  | TypesByName | All persisted types ordered by Type name |
|   | WeakReferencedTypes | Info about what Types another Type is referencing indirectly via object Id reference or other non explicit/weak reference. If your class persists objects with implicit/weak references to other persisted objects then register this fact by adding to this Dictionary(TKey, TValue) . Information provided by this Dictionary(TKey, TValue) is primarily used by Database Manager Schema Connectivity Window. |

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.Cache Property

Do cache CacheEnum.Yes

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.LookupByType Property

For use in VelocityDBExtension DLL

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Dictionary<Type, VelocityDbType> LookupByType { get; }
```

VB

```
Public ReadOnly Property LookupByType As Dictionary(Of Type, VelocityDbType)  
    Get
```

C++

```
public:  
property Dictionary<Type^, VelocityDbType^>^ LookupByType {  
    Dictionary<Type^, VelocityDbType^>^ get ();  
}
```

F#

```
member LookupByType : Dictionary<Type, VelocityDbType> with get
```

Property Value

Type: [Dictionary\(Type, VelocityDbType\)](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.TypesByName Property

All persisted types ordered by [Type](#) name

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSetOidShort<VelocityDbType> TypesByName { get; }
```

VB

```
Public ReadOnly Property TypesByName As BTreeSetOidShort(Of VelocityDbType)  
    Get
```

C++

```
public:  
property BTreeSetOidShort<VelocityDbType>^ TypesByName {  
    BTreeSetOidShort<VelocityDbType>^ get ();  
}
```

F#

```
member TypesByName : BTreeSetOidShort<VelocityDbType> with get
```

Property Value

Type: [BTreeSetOidShort\(VelocityDbType\)](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.WeakReferencedTypes Property

Info about what [Type](#) another [Type](#) is referencing indirectly via object [Id](#) reference or other non explicit/weak reference. If your class persists objects with implicit/weak references to other persisted objects then register this fact by adding to this [Dictionary\(TKey, TValue\)](#). Information provided by this [Dictionary\(TKey, TValue\)](#) is primarily used by Database Manager Schema Connectivity Window.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Dictionary<Type, List<Type>> WeakReferencedTypes { get; set; }
```

VB

```
Public Shared Property WeakReferencedTypes As Dictionary(Of Type, List(Of Type))  
    Get  
    Set
```

C++

```
public:  
static property Dictionary<Type^, List<Type^^>^ WeakReferencedTypes {  
    Dictionary<Type^, List<Type^^>^ get ();  
    void set (Dictionary<Type^, List<Type^^>^ value);  
}
```

F#

```
static member WeakReferencedTypes : Dictionary<Type, List<Type>> with get,  
set
```

Property Value

Type: [Dictionary\(Type, List\(Type\)\)](#)

See Also








[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)



Schema.Schema Methods

The [Schema](#) type exposes the following members.

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetTypeVersion | Opens a TypeVersion object |
|  InitializeAfterRecreate | This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data. (Overrides OptimizedPersistable.InitializeAfterRecreate(SessionBase) .) |
|  IsExpandedInternalType | Includes additional build in types |
|  IsInternalType(UInt16) | Determine if a given slot is within the range of slots containing the build in/pre registered types |
|  IsInternalType(VelocityDbType) | Determine if a given VelocityDbType is a build in/pre registered type |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean) .) |
|  RegisterClass | Register a type in the persistent schema |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.GetTypeVersion Method

Opens a [TypeVersion](#) object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion GetTypeVersion(  
    uint pageSlot,  
    SessionBase session,  
    bool usePersistent = false  
)
```

VB

```
Public Function GetTypeVersion (  
    pageSlot As UInteger,  
    session As SessionBase,  
    Optional usePersistent As Boolean = false  
) As TypeVersion
```

C++

```
public:  
TypeVersion^ GetTypeVersion(  
    unsigned int pageSlot,  
    SessionBase^ session,  
    bool usePersistent = false  
)
```

F#

```
member GetTypeVersion :  
    pageSlot : uint32 *  
    session : SessionBase *  
    ?usePersistent : bool  
(* Defaults:  
    let _usePersistent = defaultArg usePersistent false  
)  
-> TypeVersion
```

Parameters

pageSlot

Type: [System.UInt32](#)

Page and slot of [TypeVersion](#) object

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

VelocityDB Class Library

usePersistent (Optional)

Type: [System.Boolean](#)

Use persisted boot up type or transient one

Return Value

Type: [TypeVersion](#)

A type version

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.InitializeAfterRecreate Method

This function is called when an object has been read from disk before all data members (fields) have been fully loaded. Override this to provide your own initializations of transient data.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void InitializeAfterRecreate(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRecreate (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRecreate(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRecreate :  
    session : SessionBase -> unit  
override InitializeAfterRecreate :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session managing this object

Implements

[IOptimizedPersistable.InitializeAfterRecreate\(SessionBase\)](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.IsExpandedInternalType Method

Includes additional build in types

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsExpandedInternalType (
    ushort slotNumber
)
```

VB

```
Public Function IsExpandedInternalType (
    slotNumber As UShort
) As Boolean
```

C++

```
public:
bool IsExpandedInternalType (
    unsigned short slotNumber
)
```

F#

```
member IsExpandedInternalType :
    slotNumber : uint16 -> bool
```

Parameters

slotNumber

Type: [System.UInt16](#)

slot number of a [VelocityDbType](#) or [TypeVersion](#)

Return Value

Type: [Boolean](#)

true if internal; otherwise false



See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.IsInternalType Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|  | IsInternalType(UInt16) | Determine if a given slot is within the range of slots containing the build in/pre registered types |
|  | IsInternalType(VelocityDbType) | Determine if a given VelocityDbType is a build in/pre registered type |

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.IsInternalType Method (UInt16)

Determine if a given slot is within the range of slots containing the build in/pre registered types

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsInternalType(  
    ushort slotNumber  
)
```

VB

```
Public Function IsInternalType (  
    slotNumber As UShort  
) As Boolean
```

C++

```
public:  
bool IsInternalType(  
    unsigned short slotNumber  
)
```

F#

```
member IsInternalType :  
    slotNumber : uint16 -> bool
```

Parameters

slotNumber

Type: [System.UInt16](#)

Slot number to check

Return Value

Type: [Boolean](#)

true if slot is within the range of build in types; otherwise false

See Also

[Schema Class](#)

[IsInternalType Overload](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.IsInternalType Method (VelocityDbType)

Determine if a given [VelocityDbType](#) is a build in/pre registered type

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsInternalType(  
    VelocityDbType velocityDbType  
)
```

VB

```
Public Function IsInternalType (  
    velocityDbType As VelocityDbType  
) As Boolean
```

C++

```
public:  
bool IsInternalType(  
    VelocityDbType^ velocityDbType  
)
```

F#

```
member IsInternalType :  
    velocityDbType : VelocityDbType -> bool
```

Parameters

velocityDbType

Type: [VelocityDb.TypeInfo.VelocityDbType](#)

Return Value

Type: [Boolean](#)

true if slot is within the range of build in types; otherwise false

See Also

[Schema Class](#)

[IsInternalType Overload](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
Page^ page,  
bool useOidShort,  
Schema^ schema,  
bool openRefs,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int

32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.TypeInfo.Schema.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.RegisterClass Method

Register a type in the persistent schema

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion RegisterClass(
    Type type,
    SessionBase session,
    bool inFlush = false,
    bool persistIt = true
)
```

VB

```
Public Function RegisterClass (
    type As Type,
    session As SessionBase,
    Optional inFlush As Boolean = false,
    Optional persistIt As Boolean = true
) As TypeVersion
```

C++

```
public:
    TypeVersion^ RegisterClass(
        Type^ type,
        SessionBase^ session,
        bool inFlush = false,
        bool persistIt = true
    )
```

F#

```
member RegisterClass :
    type : Type *
    session : SessionBase *
    ?inFlush : bool *
    ?persistIt : bool
(* Defaults:
    let_inFlush = defaultArg inFlush false
    let_persistIt = defaultArg persistIt true
*)
-> TypeVersion
```

Parameters

type

Type: [System.Type](#)

A type to register

VelocityDB Class Library

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

inFlush (Optional)

Type: [System.Boolean](#)

Set to false if you want to avoid page flushing as a side affect

persistIt (Optional)

Type: [System.Boolean](#)

Set to false if getting [TypeVersion](#) for a lookup (object read)

Return Value

Type: [TypeVersion](#)

An existing [TypeVersion](#) or a new one

See Also







[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.Schema Fields

The [Schema](#) type exposes the following members.

Fields

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------|
|   | s_bootupTypeCount | VelocityDB internal schema requires this many slots on schema page |
|   | s_bootupTypeCountExpanded | Additional internal schema added Setember 9, 2017 |
|   | SchemaDB | |

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.s_bootupTypeCount Field

VelocityDB internal schema requires this many slots on schema page

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_bootupTypeCount
```

VB

```
Public Shared ReadOnly s_bootupTypeCount As UShort
```

C++

```
public:  
static initonly unsigned short s_bootupTypeCount
```

F#

```
static val s_bootupTypeCount: uint16
```

Field Value

Type: [UInt16](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.s_bootupTypeCountExpanded Field

Additional internal schema added Setember 9, 2017

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ushort s_bootupTypeCountExpanded
```

VB

```
Public Shared ReadOnly s_bootupTypeCountExpanded As UShort
```

C++

```
public:  
static initonly unsigned short s_bootupTypeCountExpanded
```

F#

```
static val s_bootupTypeCountExpanded: uint16
```

Field Value

Type: [UInt16](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

Schema.SchemaDB Field

[Missing <summary> documentation for "F:VelocityDb.TypeInfo.Schema.SchemaDB"]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const uint SchemaDB = 1
```

VB

```
Public Const SchemaDB As UInteger = 1
```

C++

```
public:  
literal unsigned int SchemaDB = 1
```

F#

```
static val mutable SchemaDB: uint32
```

Field Value

Type: [UInt32](#)

See Also

[Schema Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion Class

Contains info about a version of a [VelocityDbType](#). The info is used when reading/writing an object of the

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.TypeVersion

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class TypeVersion : OptimizedPersistable,
    IEquatable<TypeVersion>
```

VB

```
<SerializableAttribute>
Public Class TypeVersion
    Inherits OptimizedPersistable
    Implements IEquatable(Of TypeVersion)
```

C++


```
[SerializableAttribute]
public ref class TypeVersion : public OptimizedPersistable,
    IEquatable<TypeVersion^>
```
















F#

```
[<SerializableAttribute>]
type TypeVersion =
    class
        inherit OptimizedPersistable
        interface IEquatable<TypeVersion>
    end
```






The **TypeVersion** type exposes the following members.







Properties

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other |



| | | |
|-------------------------------------------------------------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | AnyFieldIndex | Does managed type or its base classes include a field index ? |
|  | AnyTypeIndex | Does managed type or its base classes include a type index ? |
|  | AutoIncrement | Does managed type or its base classes include an auto increment field? |
|  | BaseShape | Get the base TypeVersion or null if none exist |
|  | Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides OptimizedPersistable.Cache.) |
|  | CountStart | Set where AutoIncrement counter should start |
|  | DataMemberArray | Get an array of fields represented as DataMember . Does not include fields from base classes. |
|  | FieldIndex | Does managed type include a field index ? |
|  | IsFixedSize | A type is fixed size if it only has fixed size fields |
|  | IsSerializable | <code>true</code> if ISerializable otherwise, <code>false</code> . |
|  | IsString | Is this schema for type System.String ? |
|  | IsValueType | Is managed type a ValueType ? |
|  | Type | Get the .NET type of the schema class |
|  | TypeIndex | Does managed type include a type index ? |
|  | VelocityDbType | Get schema class object managing a particular .NET type. |

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  EncodeForCsv | Encode a persistent object as a string for csv export |
|  Equals | Indicates whether the current object is equal to another object of the same type. |
|  GetDataMemberList | Get a list of all DataMember of this type including base classes. |
|  LoadMembers | Loads member fields from persistent storage |
|  ObjectBytesFromStrings | Converts string representation of an object into VelocityDB object byte representation. |

| | |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  PersistReferences | Persists references from an object |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  SetCount | |
|  ToString | (Overrides OptimizedPersistable.ToString().) |
|  TotalShapeNumberOfBytes | Get the total size of an object of the managed type (including base class fields) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

















See Also

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.TypeVersion Properties

The [TypeVersion](#) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  AnyFieldIndex | Does managed type or its base classes include a field index ? |
|  AnyTypeIndex | Does managed type or its base classes include a type index ? |
|  AutoIncrement | Does managed type or its base classes include an auto increment field? |
|  BaseShape | Get the base TypeVersion or null if none exist |
|  Cache | By default, the an object cache is determined by a SessionBase constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do. (Overrides OptimizedPersistable.Cache.) |
|  CountStart | Set where AutoIncrement counter should start |
|  DataMemberArray | Get an array of fields represented as DataMember . Does not include fields from base classes. |
|  FieldIndex | Does managed type include a field index ? |
|  IsFixedSize | A type is fixed size if it only has fixed size fields |
|  IsSerializable | true if ISerializable otherwise, false. |
|  IsString | Is this schema for type System.String ? |
|  IsValueType | Is managed type a ValueType ? |
|  Type | Get the .NET type of the schema class |
|  TypeIndex | Does managed type include a type index ? |
|  VelocityDbType | Get schema class object managing a particular .NET type. |

See Also

[TypeVersion Class](#)

VelocityDB Class Library

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.AnyFieldIndex Property

Does managed type or its base classes include a field index ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AnyFieldIndex { get; }
```

VB

```
Public ReadOnly Property AnyFieldIndex As Boolean  
    Get
```

C++

```
public:  
property bool AnyFieldIndex {  
    bool get ();  
}
```

F#

```
member AnyFieldIndex : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.AnyTypeIndex Property

Does managed type or its base classes include a type index ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AnyTypeIndex { get; }
```

VB

```
Public ReadOnly Property AnyTypeIndex As Boolean  
    Get
```

C++

```
public:  
property bool AnyTypeIndex {  
    bool get ();  
}
```

F#

```
member AnyTypeIndex : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.AutoIncrement Property

Does managed type or its base classes include an auto increment field?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool AutoIncrement { get; set; }
```

VB

```
Public Property AutoIncrement As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool AutoIncrement {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member AutoIncrement : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.BaseShape Property

Get the base [TypeVersion](#) or null if none exist

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion BaseShape { get; }
```

VB

```
Public ReadOnly Property BaseShape As TypeVersion  
    Get
```

C++

```
public:  
property TypeVersion^ BaseShape {  
    TypeVersion^ get ();  
}
```

F#

```
member BaseShape : TypeVersion with get
```

Property Value

Type: [TypeVersion](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.Cache Property

By default, the an object cache is determined by a [SessionBase](#) constructor parameter but certain types of objects may be re opened more frequently than others, for such types override this to return a value. Caching objects this way can cause out of date object references to stay active due to lacking code to invalidate a cached object when referenced objects are replaced. We will add this automatic invalidation as soon as possible but for now use caution when caching objects. Caching objects that does not strongly reference other objects is OK to do.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override CacheEnum Cache { get; }
```

VB

```
Public Overrides ReadOnly Property Cache As CacheEnum  
    Get
```

C++

```
public:  
virtual property CacheEnum Cache {  
    CacheEnum get () override;  
}
```

F#

```
abstract Cache : CacheEnum with get  
override Cache : CacheEnum with get
```

Property Value

Type: [CacheEnum](#)

Implements

[IOptimizedPersistable.Cache](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.CountStart Property

Set where [AutoIncrement](#) counter should start

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ulong CountStart { get; }
```

VB

```
Public ReadOnly Property CountStart As ULong  
    Get
```

C++

```
public:  
property unsigned long long CountStart {  
    unsigned long long get ();  
}
```

F#

```
member CountStart : uint64 with get
```

Property Value

Type: [UInt64](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.DataMemberArray Property

Get an array of fields represented as [DataMember](#). Does not include fields from base classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DataMember[] DataMemberArray { get; }
```

VB

```
Public ReadOnly Property DataMemberArray As DataMember()  
    Get
```

C++

```
public:  
property array<DataMember^>^ DataMemberArray {  
    array<DataMember^>^ get ();  
}
```

F#

```
member DataMemberArray : DataMember[] with get
```

Property Value

Type: [DataMember\[\]](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.FieldIndex Property

Does managed type include a field index ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool FieldIndex { get; set; }
```

VB

```
Public Property FieldIndex As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool FieldIndex {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member FieldIndex : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.IsFixedSize Property

A type is fixed size if it only has fixed size fields

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsFixedSize { get; }
```

VB

```
Public ReadOnly Property IsFixedSize As Boolean  
    Get
```

C++

```
public:  
property bool IsFixedSize {  
    bool get ();  
}
```

F#

```
member IsFixedSize : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.IsSerializable Property

true

if [ISerializable](#) otherwise, false.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsSerializable { get; }
```

VB

```
Public ReadOnly Property IsSerializable As Boolean  
    Get
```

C++

```
public:  
property bool IsSerializable {  
    bool get ();  
}
```

F#

```
member IsSerializable : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.IsString Property

Is this schema for type System.String ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsString { get; }
```

VB

```
Public ReadOnly Property IsString As Boolean  
    Get
```

C++

```
public:  
property bool IsString {  
    bool get ();  
}
```

F#

```
member IsString : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.IsValueType Property

Is managed type a ValueType ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsValueType { get; }
```

VB

```
Public ReadOnly Property IsValueType As Boolean  
    Get
```

C++

```
public:  
property bool IsValueType {  
    bool get ();  
}
```

F#

```
member IsValueType : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.Type Property

Get the .NET type of the schema class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
property Type^ Type {  
    Type^ get ();  
}
```

F#

```
member Type : Type with get
```

Property Value

Type: [Type](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.TypeIndex Property

Does managed type include a type index ?

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool TypeIndex { get; }
```

VB

```
Public ReadOnly Property TypeIndex As Boolean  
    Get
```

C++

```
public:  
property bool TypeIndex {  
    bool get ();  
}
```

F#

```
member TypeIndex : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.VelocityDbType Property

Get schema class object managing a particular .NET type.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VelocityDbType VelocityDbType { get; }
```

VB

```
Public ReadOnly Property VelocityDbType As VelocityDbType  
    Get
```

C++

```
public:  
property VelocityDbType^ VelocityDbType {  
    VelocityDbType^ get ();  
}
```

F#

```
member VelocityDbType : VelocityDbType with get
```

Property Value

Type: [VelocityDbType](#)

See Also












[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)



TypeVersion.TypeVersion Methods

The [TypeVersion](#) type exposes the following members.

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  EncodeForCsv | Encode a persistent object as a string for csv export |
|  Equals | Indicates whether the current object is equal to another object of the same type. |
|  GetDataMemberList | Get a list of all DataMember of this type including base classes. |
|  LoadMembers | Loads member fields from persistent storage |
|  ObjectBytesFromStrings | Converts string representation of an object into VelocityDB object byte representation. |
|  PersistRefences | Persists references from an object |
|  ReadMe | (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  SetCount | |
|  ToString | (Overrides OptimizedPersistable.ToString().) |
|  TotalShapeNumberOfBytes | Get the total size of an object of the managed type (including base class fields) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.EncodeForCsv Method

Encode a persistent object as a string for csv export

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EncodeForCsv(  
    Object obj,  
    PageInfo pageInfo,  
    SessionBase session  
)
```

VB

```
Public Function EncodeForCsv (  
    obj As Object,  
    pageInfo As PageInfo,  
    session As SessionBase  
) As String
```

C++

```
public:  
String^ EncodeForCsv(  
    Object^ obj,  
    PageInfo^ pageInfo,  
    SessionBase^ session  
)
```

F#

```
member EncodeForCsv :  
    obj : Object *  
    pageInfo : PageInfo *  
    session : SessionBase -> string
```

Parameters

obj

Type: [System.Object](#)

Object to be encoded

pageInfo

Type: [VelocityDb.PageInfo](#)

PageInfo for page where object lives

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

VelocityDB Class Library

Return Value

Type: [String](#)

Object encoded as a string

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.Equals Method

Indicates whether the current object is equal to another object of the same type.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    TypeVersion other  
)
```

VB

```
Public Function Equals (  
    other As TypeVersion  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    TypeVersion^ other  
) sealed
```

F#

```
abstract Equals :  
    other : TypeVersion -> bool  
override Equals :  
    other : TypeVersion -> bool
```

Parameters

other

Type: [VelocityDb.TypeInfo.TypeVersion](#)

An object to compare with this object.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if the current object is equal to the *other* parameter; otherwise, `false` (`False` in Visual Basic).

Implements

[IEquatable\(T\).Equals\(T\)](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.GetDataMemberList Method

Get a list of all [DataMember](#) of this type including base classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<DataMember> GetDataMemberList ()
```

VB

```
Public Function GetDataMemberList As List(Of DataMember)
```

C++

```
public:  
List<DataMember^>^ GetDataMemberList ()
```

F#

```
member GetDataMemberList : unit -> List<DataMember>
```

Return Value

Type: [List\(DataMember\)](#)

A list of [DataMember](#) that this type or a base class contains

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.LoadMembers Method

Loads member fields from persistent storage

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void LoadMembers(  
    IOptimizedPersistable toLoad,  
    Schema schema,  
    List<IOptimizedPersistable> toLoadMembers,  
    int graphDepth,  
    int graphDepthToLoad,  
    SessionBase session,  
    bool primitivesOnly = false  
)
```

VB

```
Public Sub LoadMembers (  
    toLoad As IOptimizedPersistable,  
    schema As Schema,  
    toLoadMembers As List(Of IOptimizedPersistable),  
    graphDepth As Integer,  
    graphDepthToLoad As Integer,  
    session As SessionBase,  
    Optional primitivesOnly As Boolean = false  
)
```

C++

```
public:  
void LoadMembers (  
    IOptimizedPersistable^ toLoad,  
    Schema^ schema,  
    List<IOptimizedPersistable^>^ toLoadMembers,  
    int graphDepth,  
    int graphDepthToLoad,  
    SessionBase^ session,  
    bool primitivesOnly = false  
)
```

F#

```
member LoadMembers :  
    toLoad : IOptimizedPersistable *  
    schema : Schema *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    session : SessionBase *  
    ?primitivesOnly : bool
```

VelocityDB Class Library

```
(* Defaults:  
    let_primitivesOnly = defaultArg primitivesOnly false  
*)  
-> unit
```

Parameters

toLoad

Type: [VelocityDb.IOptimizedPersistable](#)

Object to load fields for

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The current schema

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

To be loaded fields list

graphDepth

Type: [System.Int32](#)

Current graph depth loaded

graphDepthToLoad

Type: [System.Int32](#)

Max graph depth to load fields for

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

primitivesOnly (Optional)

Type: [System.Boolean](#)

If true, load only primitive values

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.ObjectBytesFromStrings Method

Converts string representation of an object into VelocityDB object byte representation.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ObjectBytesFromStrings (
    string[] members,
    string[] fieldNames,
    SessionBase session,
    Schema schema
)
```

VB

```
Public Sub ObjectBytesFromStrings (
    members As String(),
    fieldNames As String(),
    session As SessionBase,
    schema As Schema
)
```

C++

```
public:
void ObjectBytesFromStrings (
    array<String^>^ members,
    array<String^>^ fieldNames,
    SessionBase^ session,
    Schema^ schema
)
```

F#

```
member ObjectBytesFromStrings :
    members : string[] *
    fieldNames : string[] *
    session : SessionBase *
    schema : Schema -> unit
```

Parameters

members

Type: [System.String\[\]](#)

fieldNames

Type: [System.String\[\]](#)

session

Type: [VelocityDb.Session.SessionBase](#)

Active session

VelocityDB Class Library

schema

Type: [VelocityDb.TypeInfo.Schema](#)

Current schema

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.PersistReferences Method

Persists references from an object

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void PersistReferences (
    Object obj,
    PageInfo pageInfo,
    IOptimizedPersistable owner,
    SessionBase session,
    bool inFlush
)
```

VB

```
Public Sub PersistReferences (
    obj As Object,
    pageInfo As PageInfo,
    owner As IOptimizedPersistable,
    session As SessionBase,
    inFlush As Boolean
)
```

C++

```
public:
void PersistReferences (
    Object^ obj,
    PageInfo^ pageInfo,
    IOptimizedPersistable^ owner,
    SessionBase^ session,
    bool inFlush
)
```

F#

```
member PersistReferences :
    obj : Object *
    pageInfo : PageInfo *
    owner : IOptimizedPersistable *
    session : SessionBase *
    inFlush : bool -> unit
```

Parameters

obj

Type: [System.Object](#)

Object for which we want to persist all references

VelocityDB Class Library

pageInfo

Type: [VelocityDb.PageInfo](#)

PageInfo for page containing the object for which we are persisting references

owner

Type: [VelocityDb.IOptimizedPersistable](#)

Owner object of the object

session

Type: [VelocityDb.Session.SessionBase](#)

Session managing the object

inFlush

Type: [System.Boolean](#)

Are we currently within a page flush

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.ReadMe Method

[Missing <summary> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Boolean)"]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
```

```
Page^ page,  
bool useOidShort,  
Schema^ schema,  
bool openRefs,  
List<IOptimizedPersistable^>^ toLoadMembers,  
int graphDepth,  
int graphDepthToLoad,  
bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

[Missing <param name="typeVersion"/> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

memberBytes

Type: [System.Byte\[\]](#)

[Missing <param name="memberBytes"/> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System

`m.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]`

offset

Type: [System.Int32](#)

[Missing <param name="offset"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

useOidShort

Type: [System.Boolean](#)

[Missing <param name="useOidShort"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

schema

Type: [VelocityDb.TypeInfo.Schema](#)

[Missing <param name="schema"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

openRefs

Type: [System.Boolean](#)

[Missing <param name="openRefs"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

[Missing <param name="toLoadMembers"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepth

Type: [System.Int32](#)

[Missing <param name="graphDepth"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

graphDepthToLoad

Type: [System.Int32](#)

[Missing <param name="graphDepthToLoad"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

primitivesOnly

Type: [System.Boolean](#)

[Missing <param name="primitivesOnly"/> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ReadMe(VelocityDb.TypeInfo.TypeVersion,System.Byte[],System.Int32@,VelocityDb.Session.SessionBase,VelocityDb.Page,System.Boolean,VelocityDb.TypeInfo.Schema,System.Boolean,System.Collections.Generic.List{VelocityDb.IOptimizedPersistable},System.Int32,System.Int32,System.Boolean)"]

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.SetCount Method

[Missing <summary> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.SetCount(System.Object,VelocityDb.AutoPlacement)"]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetCount (
    Object obj,
    AutoPlacement autoPlacement
)
```

VB

```
Public Sub SetCount (
    obj As Object,
    autoPlacement As AutoPlacement
)
```

C++

```
public:
void SetCount (
    Object^ obj,
    AutoPlacement^ autoPlacement
)
```

F#

```
member SetCount :
    obj : Object *
    autoPlacement : AutoPlacement -> unit
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.SetCount(System.Object,VelocityDb.AutoPlacement)"]

autoPlacement

Type: [VelocityDb.AutoPlacement](#)

[Missing <param name="autoPlacement"/> documentation for

"M:VelocityDb.TypeInfo.TypeVersion.SetCount(System.Object,VelocityDb.AutoPlacement)"]

See Also

[TypeVersion Class](#)

VelocityDB Class Library

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.ToString Method

[Missing <summary> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ToString"]

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDb.TypeInfo.TypeVersion.ToString"]

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.TotalShapeNumberOfBytes Method

Get the total size of an object of the managed type (including base class fields)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int TotalShapeNumberOfBytes (
    Schema schema
)
```

VB

```
Public Function TotalShapeNumberOfBytes (
    schema As Schema
) As Integer
```

C++

```
public:
int TotalShapeNumberOfBytes (
    Schema^ schema
)
```

F#

```
member TotalShapeNumberOfBytes :
    schema : Schema -> int
```

Parameters

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active schema

Return Value

Type: [Int32](#)

The total size of all fields or -1 if fields do not have a fixed size

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

TypeVersion.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[TypeVersion Class](#)

[VelocityDb.TypeInfo Namespace](#)

UseOidShort Class

Indicates that every field or selected fields of a serializable class should be referenced using a `OidShort` instead of a full `Oid`. This means that the referenced object is located in the same database as the object containing the reference. A `ShortOid` uses less space and references are not tied to a certain database number.

Inheritance Hierarchy

[System.Object](#)

[System.Attribute](#)

VelocityDb.TypeInfo.UseOidShort

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public sealed class UseOidShort : Attribute
```

VB

```
Public NotInheritable Class UseOidShort  
    Inherits Attribute
```

C++


```
public ref class UseOidShort sealed : public Attribute
```

F#

```
[<SealedAttribute>]  
type UseOidShort =  
    class  
        inherit Attribute  
    end
```

The **UseOidShort** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------|
|  | UseOidShort | Creates the attribute indicating use of <code>OidShort</code> (page + page) |

See Also

[VelocityDb.TypeInfo Namespace](#)

UseOidShort Constructor

Creates the attribute indicating use of OidShort (page + page)

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public UseOidShort ()
```

VB

```
Public Sub New
```

C++

```
public:  
UseOidShort ()
```

F#

```
new : unit -> UseOidShort
```

See Also

[UseOidShort Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType Class

Contains persistent info for a [Type](#)

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDb.TypeInfo.VelocityDbType

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class VelocityDbType : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class VelocityDbType
    Inherits OptimizedPersistable
```

C++




```
[SerializableAttribute]
public ref class VelocityDbType : public OptimizedPersistable
```


F#

```
[<SerializableAttribute>]
type VelocityDbType =
    class
        inherit OptimizedPersistable
    end
```





The **VelocityDbType** type exposes the following members.

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage.) |
|  | Type | Get the type of the schema managed class |
|  | TypeName | Name of Type that this meta object is managing. |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  | TypeVersions | Initially one version exist, if class is modified and UpdateClass(Type) is called other versions can exist. |
|-----------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------------------------------------------------------|

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | LastShape | Get the latest version of a schema class |
|  | ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[], Int32, SessionBase, Page, Boolean, Schema, Boolean, List(IOptimizedPersistable), Int32, Int32, Boolean).) |
|  | ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString().) |
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |





See Also

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.VelocityDbType Properties

The [VelocityDbType](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AllowOtherTypesOnSamePage | Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types. (Overrides OptimizedPersistable.AllowOtherTypesOnSamePage .) |
|  | Type | Get the type of the schema managed class |
|  | TypeName | Name of Type that this meta object is managing. |
|  | TypeVersions | Initially one version exist, if class is modified and UpdateClass(Type) is called other versions can exist. |

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.AllowOtherTypesOnSamePage Property

Objects can be stored more efficiently if all object types on the page share the same type. By default mixed types are allowed. Override this to return false for types that should not share pages with other types.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool AllowOtherTypesOnSamePage { get; }
```

VB

```
Public Overrides ReadOnly Property AllowOtherTypesOnSamePage As Boolean  
    Get
```

C++

```
public:  
virtual property bool AllowOtherTypesOnSamePage {  
    bool get () override;  
}
```

F#

```
abstract AllowOtherTypesOnSamePage : bool with get  
override AllowOtherTypesOnSamePage : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IOptimizedPersistable.AllowOtherTypesOnSamePage](#)

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.Type Property

Get the type of the schema managed class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
property Type^ Type {  
    Type^ get ();  
}
```

F#

```
member Type : Type with get
```

Property Value

Type: [Type](#)

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.TypeName Property

Name of [Type](#) that this meta object is managing.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TypeName { get; }
```

VB

```
Public ReadOnly Property TypeName As String  
    Get
```

C++

```
public:  
property String^ TypeName {  
    String^ get ();  
}
```

F#

```
member TypeName : string with get
```

Property Value

Type: [String](#)

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.TypeVersions Property

Initially one version exist, if class is modified and [UpdateClass\(Type\)](#) is called other versions can exist.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion[] TypeVersions { get; }
```

VB

```
Public ReadOnly Property TypeVersions As TypeVersion()  
    Get
```

C++

```
public:  
property array<TypeVersion^>^ TypeVersions {  
    array<TypeVersion^>^ get ();  
}
```

F#

```
member TypeVersions : TypeVersion[] with get
```

Property Value

Type: [TypeVersion\[\]](#)

See Also





[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)



VelocityDbType.VelocityDbType Methods

The [VelocityDbType](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  LastShape | Get the latest version of a schema class |
|  ReadMe | Used by code generator (Overrides OptimizedPersistable.ReadMe(TypeVersion,Byte[],Int32,SessionBase,Page,Boolean,Schema,Boolean,List(IOptimizedPersistable),Int32,Int32,Boolean).) |
|  ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString().) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.LastShape Method

Get the latest version of a schema class

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TypeVersion LastShape ()
```

VB

```
Public Function LastShape As TypeVersion
```

C++

```
public:  
TypeVersion^ LastShape ()
```

F#

```
member LastShape : unit -> TypeVersion
```

Return Value

Type: [TypeVersion](#)

Current version of a schema class

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.ReadMe Method

Used by code generator

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void ReadMe (
    TypeVersion typeVersion,
    byte[] memberBytes,
    ref int offset,
    SessionBase session,
    Page page,
    bool useOidShort,
    Schema schema,
    bool openRefs,
    List<IOptimizedPersistable> toLoadMembers,
    int graphDepth,
    int graphDepthToLoad,
    bool primitivesOnly
)
```

VB

```
Public Overrides Sub ReadMe (
    typeVersion As TypeVersion,
    memberBytes As Byte(),
    ByRef offset As Integer,
    session As SessionBase,
    page As Page,
    useOidShort As Boolean,
    schema As Schema,
    openRefs As Boolean,
    toLoadMembers As List(Of IOptimizedPersistable),
    graphDepth As Integer,
    graphDepthToLoad As Integer,
    primitivesOnly As Boolean
)
```

C++

```
public:
virtual void ReadMe (
    TypeVersion^ typeVersion,
    array<unsigned char>^ memberBytes,
    int% offset,
    SessionBase^ session,
    Page^ page,
    bool useOidShort,
    Schema^ schema,
    bool openRefs,
    List<IOptimizedPersistable^>^ toLoadMembers,
```

VelocityDB Class Library

```
    int graphDepth,  
    int graphDepthToLoad,  
    bool primitivesOnly  
) override
```

F#

```
abstract ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit  
override ReadMe :  
    typeVersion : TypeVersion *  
    memberBytes : byte[] *  
    offset : int byref *  
    session : SessionBase *  
    page : Page *  
    useOidShort : bool *  
    schema : Schema *  
    openRefs : bool *  
    toLoadMembers : List<IOptimizedPersistable> *  
    graphDepth : int *  
    graphDepthToLoad : int *  
    primitivesOnly : bool -> unit
```

Parameters

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

The version of the type being read

memberBytes

Type: [System.Byte\[\]](#)

offset

Type: [System.Int32](#)

session

Type: [VelocityDb.Session.SessionBase](#)

page

Type: [VelocityDb.Page](#)

useOidShort

Type: [System.Boolean](#)

schema

Type: [VelocityDb.TypeInfo.Schema](#)

VelocityDB Class Library

openRefs

Type: [System.Boolean](#)

toLoadMembers

Type: [System.Collections.Generic.List<IOptimizedPersistable>](#)

graphDepth

Type: [System.Int32](#)

graphDepthToLoad

Type: [System.Int32](#)

primitivesOnly

Type: [System.Boolean](#)

Implements

[IOptimizedPersistable.ReadMe\(TypeVersion,Byte\[\], Int32, SessionBase, Page, Boolean, Schema, Boolean, List<IOptimizedPersistable>, Int32, Int32, Boolean\)](#)

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.ToString Method

Displays class name plus object id

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDbType.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityDb.TypeInfo](#)

Assembly: VelocityDb (in VelocityDb.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also







[VelocityDbType Class](#)

[VelocityDb.TypeInfo Namespace](#)

VelocityDBExtensions Namespace

The **VelocityDBExtensions** namespace contains extensions to VelocityDB.

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | FieldsOnlyContractResolver | |
|  | ImportExportCsv | |
|  | JsonImportExport | |
|  | Sync | |
|  | SyncProvider | VelocityDB standard Microsoft Sync provider as described in https://msdn.microsoft.com/en-us/library/bb902826(v=sql.110).aspx |
|  | Utilities | Currently only used by Database Manager |

FieldsOnlyContractResolver Class

[Missing <summary> documentation for "T:VelocityDBExtensions.FieldsOnlyContractResolver"]

Inheritance Hierarchy

[System.Object](#)

DefaultContractResolver

VelocityDBExtensions.FieldsOnlyContractResolver

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class FieldsOnlyContractResolver : DefaultContractResolver
```

VB

```
Public Class FieldsOnlyContractResolver  
    Inherits DefaultContractResolver
```

C++


```
public ref class FieldsOnlyContractResolver : public DefaultContractResolver
```

F#

```
type FieldsOnlyContractResolver =  
    class  
        inherit DefaultContractResolver  
    end
```

The **FieldsOnlyContractResolver** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------|
|  | FieldsOnlyContractResolver | Initializes a new instance of the FieldsOnlyContractResolver class |

See Also

[VelocityDBExtensions Namespace](#)

FieldsOnlyContractResolver Constructor

Initializes a new instance of the [FieldsOnlyContractResolver](#) class

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public FieldsOnlyContractResolver ()
```

VB

```
Public Sub New
```

C++

```
public:  
FieldsOnlyContractResolver ()
```

F#

```
new : unit -> FieldsOnlyContractResolver
```

See Also

[FieldsOnlyContractResolver Class](#)

[VelocityDBExtensions Namespace](#)

ImportExportCsv Class

[Missing <summary> documentation for "T:VelocityDBExtensions.ImportExportCsv"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.ImportExportCsv

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class ImportExportCsv
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class ImportExportCsv
```

C++



```
[ExtensionAttribute]  
public ref class ImportExportCsv abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type ImportExportCsv = class end
```

The **ImportExportCsv** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV |



See Also

[VelocityDBExtensions Namespace](#)

ImportExportCsv.ImportExportCsv Methods

The [ImportExportCsv](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx |
|  | ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV |

See Also

[ImportExportCsv Class](#)

[VelocityDBExtensions Namespace](#)

ImportExportCsv.ExportToCSV Method

Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded [http://msdn.microsoft.com/en-us/library/dhx0d524\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx)

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static void ExportToCSV(  
    this SessionBase session,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub ExportToCSV (  
    session As SessionBase,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void ExportToCSV(  
    SessionBase^ session,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member ExportToCSV :  
    session : SessionBase *  
    directory : string -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

the active session

directory

Type: [System.String](#)

Where to store the CSV files

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ImportExportCsv Class](#)

[VelocityDBExtensions Namespace](#)

ImportExportCsv.ImportFromCSV Method

Restores database files, pages and objects from a .csv file data created with ExportToCSV

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static void ImportFromCSV(  
    this SessionBase session,  
    string csvDirectory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub ImportFromCSV (  
    session As SessionBase,  
    csvDirectory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void ImportFromCSV(  
    SessionBase^ session,  
    String^ csvDirectory  
)
```

F#

```
[<ExtensionAttribute>]  
static member ImportFromCSV :  
    session : SessionBase *  
    csvDirectory : string -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

the active session

csvDirectory

Type: [System.String](#)

Path to directory containing CSV files

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ImportExportCsv Class](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport Class

[Missing <summary> documentation for "T:VelocityDBExtensions.JsonImportExport"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.JsonImportExport

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class JsonImportExport
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class JsonImportExport
```

C++





```
[ExtensionAttribute]  
public ref class JsonImportExport abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type JsonImportExport = class end
```

The **JsonImportExport** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------|-------------|
|  | ExportToJson(T)(SessionBase, UInt64) | |
|  | ExportToJson(T)(SessionBase, Oid) | |
|  | ExportToJson(T)(SessionBase, Boolean, Boolean) | |
|  | ImportJson(T) | |





See Also

[VelocityDBExtensions Namespace](#)

JsonImportExport.JsonImportExport Methods

The [JsonImportExport](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|-------------|
|  | ExportToJson(T)(SessionBase, UInt64) | |
|  | ExportToJson(T)(SessionBase, Oid) | |
|  | ExportToJson(T)(SessionBase, Boolean, Boolean) | |
|  | ImportJson(T) | |




See Also

[JsonImportExport Class](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport.ExportToJson Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|-------------|
|  | ExportToJson(T)(SessionBase, UInt64) | |
|  | ExportToJson(T)(SessionBase, Oid) | |
|  | ExportToJson(T)(SessionBase, Boolean, Boolean) | |

See Also

[JsonImportExport Class](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport.ExportToJson(*T*) Method (SessionBase, UInt64)

[Missing <summary> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.UInt64)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ExportToJson<T>(
    this SessionBase session,
    ulong id
)
```

VB

```
<ExtensionAttribute>
Public Shared Function ExportToJson(Of T) (
    session As SessionBase,
    id As ULong
) As String
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
static String^ ExportToJson(
    SessionBase^ session,
    unsigned long long id
)
```

F#

```
[<ExtensionAttribute>]
static member ExportToJson :
    session : SessionBase *
    id : uint64 -> string
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.UInt64)"]

id

Type: [System.UInt64](#)

[Missing <param name="id"/> documentation for
"M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.
UInt64)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for
"M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.
UInt64)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for
"M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.
UInt64)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[JsonImportExport Class](#)

[ExportToJson Overload](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport.ExportToJson(T) Method (SessionBase, Oid)

[Missing <summary> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,VelocityDb.Oid)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ExportToJson<T>(
    this SessionBase session,
    Oid oid
)
```

VB

```
<ExtensionAttribute>
Public Shared Function ExportToJson(Of T) (
    session As SessionBase,
    oid As Oid
) As String
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
static String^ ExportToJson(
    SessionBase^ session,
    Oid oid
)
```

F#

```
[<ExtensionAttribute>]
static member ExportToJson :
    session : SessionBase *
    oid : Oid -> string
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,VelocityDb.Oid)"]

oid

Type: [VelocityDb.Oid](#)

[Missing <param name="oid"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,VelocityDb.Oid)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,VelocityDb.Oid)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,VelocityDb.Oid)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[JsonImportExport Class](#)

[ExportToJson Overload](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport.ExportToJson(*T*) Method (SessionBase, Boolean, Boolean)

[Missing <summary> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static IEnumerable<string> ExportToJson<T>(
    this SessionBase session,
    bool includeSubclasses = false,
    bool databasePerType = true
)
```

VB

```
<ExtensionAttribute>
Public Shared Function ExportToJson(Of T) (
    session As SessionBase,
    Optional includeSubclasses As Boolean = false,
    Optional databasePerType As Boolean = true
) As IEnumerable(Of String)
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
static IEnumerable<String^> ExportToJson(
    SessionBase^ session,
    bool includeSubclasses = false,
    bool databasePerType = true
)
```

F#

```
[<ExtensionAttribute>]
static member ExportToJson :
    session : SessionBase *
    ?includeSubclasses : bool *
    ?databasePerType : bool
(* Defaults:
    let _includeSubclasses = defaultArg includeSubclasses false
    let _databasePerType = defaultArg databasePerType true
*)
-> IEnumerable<string>
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

includeSubclasses (Optional)

Type: [System.Boolean](#)

[Missing <param name="includeSubclasses"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

databasePerType (Optional)

Type: [System.Boolean](#)

[Missing <param name="databasePerType"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.JsonImportExport.ExportToJson`1(VelocityDb.Session.SessionBase,System.Boolean,System.Boolean)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[JsonImportExport Class](#)

[ExportToJson Overload](#)

[VelocityDBExtensions Namespace](#)

JsonImportExport.ImportJson(T) Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.JsonImportExport.ImportJson`1(VelocityDb.Session.SessionBase,System.String)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static T ImportJson<T>(
    this SessionBase session,
    string json
)
```

VB

```
<ExtensionAttribute>
Public Shared Function ImportJson(Of T) (
    session As SessionBase,
    json As String
) As T
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
static T ImportJson(
    SessionBase^ session,
    String^ json
)
```

F#

```
[<ExtensionAttribute>]
static member ImportJson :
    session : SessionBase *
    json : string -> 'T
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDBExtensions.JsonImportExport.ImportJson`1(VelocityDb.Session.SessionBase,System.String)"]

json

Type: [System.String](#)

[Missing <param name="json"/> documentation for

"M:VelocityDBExtensions.JsonImportExport.ImportJson`1(VelocityDb.Session.SessionBase,System.String)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for

"M:VelocityDBExtensions.JsonImportExport.ImportJson`1(VelocityDb.Session.SessionBase,System.String)"]

Return Value

Type: **T**

[Missing <returns> documentation for

"M:VelocityDBExtensions.JsonImportExport.ImportJson`1(VelocityDb.Session.SessionBase,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[JsonImportExport Class](#)

[VelocityDBExtensions Namespace](#)

Sync Class

[Missing <summary> documentation for "T:VelocityDBExtensions.Sync"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Sync

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class Sync
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class Sync
```

C++




```
[ExtensionAttribute]  
public ref class Sync abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type Sync = class end
```

The **Sync** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-------------|
|  | MicrosoftSync | |
|  | SyncWith(SessionBase, SessionBase) | |
|  | SyncWith(SessionBase, SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | |




See Also

[VelocityDBExtensions Namespace](#)

Sync.Sync Methods

The [Sync](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-------------|
|  | MicrosoftSync | |
|  | SyncWith(SessionBase, SessionBase) | |
|  | SyncWith(SessionBase, SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | |

See Also

[Sync Class](#)

[VelocityDBExtensions Namespace](#)

Sync.MicrosoftSync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Sync.MicrosoftSync(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static SyncOperationStatistics MicrosoftSync(  
    this SessionBase sessionToUpdate,  
    SessionBase sessionToRead  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function MicrosoftSync (  
    sessionToUpdate As SessionBase,  
    sessionToRead As SessionBase  
) As SyncOperationStatistics
```

C++

```
public:  
[ExtensionAttribute]  
static SyncOperationStatistics^ MicrosoftSync(  
    SessionBase^ sessionToUpdate,  
    SessionBase^ sessionToRead  
)
```

F#

```
[<ExtensionAttribute>]  
static member MicrosoftSync :  
    sessionToUpdate : SessionBase *  
    sessionToRead : SessionBase -> SyncOperationStatistics
```

Parameters

sessionToUpdate

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionToUpdate"/> documentation for

"M:VelocityDBExtensions.Sync.MicrosoftSync(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

sessionToRead

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionToRead"/> documentation for "M:VelocityDBExtensions.Sync.MicrosoftSync(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

Return Value

Type: **SyncOperationStatistics**

[Missing <returns> documentation for "M:VelocityDBExtensions.Sync.MicrosoftSync(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).



See Also

[Sync Class](#)

[VelocityDBExtensions Namespace](#)

Sync.SyncWith Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-------------|
|  | SyncWith(SessionBase, SessionBase) | |
|  | SyncWith(SessionBase, SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | |

See Also

[Sync Class](#)

[VelocityDBExtensions Namespace](#)

Sync.SyncWith Method (SessionBase, SessionBase)

[Missing <summary> documentation for

"M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static void SyncWith(  
    this SessionBase sessionToUpdate,  
    SessionBase sessionOther  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SyncWith (  
    sessionToUpdate As SessionBase,  
    sessionOther As SessionBase  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SyncWith(  
    SessionBase^ sessionToUpdate,  
    SessionBase^ sessionOther  
)
```

F#

```
[<ExtensionAttribute>]  
static member SyncWith :  
    sessionToUpdate : SessionBase *  
    sessionOther : SessionBase -> unit
```

Parameters

sessionToUpdate

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionToUpdate"/> documentation for

"M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

sessionOther

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionOther"/> documentation for "M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[Sync Class](#)

[SyncWith Overload](#)

[VelocityDBExtensions Namespace](#)

Sync.SyncWith Method (SessionBase, SessionBase, Func(SessionBase, UInt64, Change, Boolean))

[Missing <summary> documentation for

"M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase,System.Func{VelocityDb.Session.SessionBase,System.UInt64,VelocityDb.Sync.Change,System.Boolean})"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static void SyncWith(  
    this SessionBase sessionToUpdate,  
    SessionBase sessionToRead,  
    Func<SessionBase, ulong, Change, bool> doUpdate  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SyncWith (  
    sessionToUpdate As SessionBase,  
    sessionToRead As SessionBase,  
    doUpdate As Func(Of SessionBase, ULong, Change, Boolean)  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SyncWith(  
    SessionBase^ sessionToUpdate,  
    SessionBase^ sessionToRead,  
    Func<SessionBase^, unsigned long long, Change^, bool>^ doUpdate  
)
```

F#

```
[<ExtensionAttribute>]  
static member SyncWith :  
    sessionToUpdate : SessionBase *  
    sessionToRead : SessionBase *  
    doUpdate : Func<SessionBase, uint64, Change, bool> -> unit
```

Parameters

sessionToUpdate

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionToUpdate"/> documentation for "M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase,System.Func{VelocityDb.Session.SessionBase,System.UInt64,VelocityDb.Sync.Change,System.Boolean})"]

sessionToRead

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="sessionToRead"/> documentation for "M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase,System.Func{VelocityDb.Session.SessionBase,System.UInt64,VelocityDb.Sync.Change,System.Boolean})"]

doUpdate

Type: [System.Func\(SessionBase, UInt64, Change, Boolean\)](#)

[Missing <param name="doUpdate"/> documentation for "M:VelocityDBExtensions.Sync.SyncWith(VelocityDb.Session.SessionBase,VelocityDb.Session.SessionBase,System.Func{VelocityDb.Session.SessionBase,System.UInt64,VelocityDb.Sync.Change,System.Boolean})"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [SessionBase](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[Sync Class](#)

[SyncWith Overload](#)

[VelocityDBExtensions Namespace](#)

SyncProvider Class

VelocityDB standard Microsoft Sync provider as described in [https://msdn.microsoft.com/en-us/library/bb902826\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/bb902826(v=sql.110).aspx)

Inheritance Hierarchy

[System.Object](#)

SyncProvider

KnowledgeSyncProvider

VelocityDBExtensions.SyncProvider

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class SyncProvider : KnowledgeSyncProvider,  
    IChangeDataRetriever, INotifyingChangeApplierTarget
```

VB

```
Public Class SyncProvider  
    Inherits KnowledgeSyncProvider  
    Implements IChangeDataRetriever, INotifyingChangeApplierTarget
```

C++


```
public ref class SyncProvider : public KnowledgeSyncProvider,  
    IChangeDataRetriever, INotifyingChangeApplierTarget
```

F#


```
type SyncProvider =  
    class  
        inherit KnowledgeSyncProvider  
        interface IChangeDataRetriever  
        interface INotifyingChangeApplierTarget  
    end
```

The **SyncProvider** type exposes the following members.

















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | SyncProvider | Initializes a new instance of the SyncProvider class |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|------------------------------------------------|
|  | IdFormats | (Overrides KnowledgeSyncProvider.IdFormats().) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | BeginSession | (Overrides KnowledgeSyncProvider.BeginSession(SyncProviderPosition, SyncSessionContext).) |
|  | EndSession | (Overrides KnowledgeSyncProvider.EndSession(SyncSessionContext).) |
|  | Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|  | GetChangeBatch | (Overrides KnowledgeSyncProvider.GetChangeBatch(UInt32, SyncKnowledge, Object).) |
|  | GetDataRetriever | |
|  | GetFullEnumerationChangeBatch | (Overrides KnowledgeSyncProvider.GetFullEnumerationChangeBatch(UInt32, SyncId, SyncKnowledge, Object).) |
|  | GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  | GetNextTickCount | |
|  | GetSyncBatchParameters | (Overrides KnowledgeSyncProvider.GetSyncBatchParameters(UInt32, SyncKnowledge).) |
|  | LoadChangeData | |
|  | ProcessChangeBatch | (Overrides KnowledgeSyncProvider.ProcessChangeBatch(ConflictResolutionPolicy, ChangeBatch, Object, SyncCallbacks, SyncSessionStatistics).) |
|  | ProcessFullEnumerationChangeBatch | (Overrides KnowledgeSyncProvider.ProcessFullEnumerationChangeBatch(ConflictResolutionPolicy, FullEnumerationChangeBatch, Object, SyncCallbacks, SyncSessionStatistics).) |
|  | SaveChangeWithChangeUnits | |
|  | SaveConflict | |
|  | SaveItemChange | |
|  | StoreKnowledgeForScope | |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------|--|
|  TryGetDestinationVersion | |
|----------------------------------------------------------------------------------------------------------------------------|--|

See Also

[VelocityDBExtensions Namespace](#)

SyncProvider Constructor

Initializes a new instance of the [SyncProvider](#) class

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public SyncProvider(  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    session As SessionBase  
)
```

C++

```
public:  
SyncProvider(  
    SessionBase^ session  
)
```

F#

```
new :  
    session : SessionBase -> SyncProvider
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions.SyncProvider.#ctor(VelocityDb.Session.SessionBase)"]

See Also


[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.SyncProvider Properties

The [SyncProvider](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|------------------------------------------------|
|  | IdFormats | (Overrides KnowledgeSyncProvider.IdFormats().) |

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.IdFormats Property

[Missing <summary> documentation for "P:VelocityDBExtensions.SyncProvider.IdFormats"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override SyncIdFormatGroup IdFormats { get; }
```

VB

```
Public Overrides ReadOnly Property IdFormats As SyncIdFormatGroup  
    Get
```

C++

```
public:  
virtual property SyncIdFormatGroup^ IdFormats {  
    SyncIdFormatGroup^ get () override;  
}
```

F#

```
abstract IdFormats : SyncIdFormatGroup with get  
override IdFormats : SyncIdFormatGroup with get
```

Property Value

Type: **SyncIdFormatGroup**

Implements

[IChangeDataRetriever.IdFormats\(\)](#)

[INotifyingChangeApplierTarget.IdFormats\(\)](#)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.SyncProvider Methods

The [SyncProvider](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BeginSession | (Overrides KnowledgeSyncProvider.BeginSession(SyncProviderPosition, SyncSessionContext).) |
| EndSession | (Overrides KnowledgeSyncProvider.EndSession(SyncSessionContext).) |
| Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object).) |
| GetChangeBatch | (Overrides KnowledgeSyncProvider.GetChangeBatch(UInt32, SyncKnowledge, Object).) |
| GetDataRetriever | |
| GetFullEnumerationChangeBatch | (Overrides KnowledgeSyncProvider.GetFullEnumerationChangeBatch(UInt32, SyncId, SyncKnowledge, Object).) |
| GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode().) |
| GetNextTickCount | |
| GetSyncBatchParameters | (Overrides KnowledgeSyncProvider.GetSyncBatchParameters(UInt32, SyncKnowledge).) |
| LoadChangeData | |
| ProcessChangeBatch | (Overrides KnowledgeSyncProvider.ProcessChangeBatch(ConflictResolutionPolicy, ChangeBatch, Object, SyncCallbacks, SyncSessionStatistics).) |
| ProcessFullEnumerationChangeBatch | (Overrides KnowledgeSyncProvider.ProcessFullEnumerationChangeBatch(ConflictResolutionPolicy, FullEnumerationChangeBatch, Object, SyncCallbacks, SyncSessionStatistics).) |
| SaveChangeWithChangeUnits | |
| SaveConflict | |
| SaveItemChange | |
| StoreKnowledgeForScope | |
| ToString | Returns a string that represents the current object. (Overrides Object.ToString().) |
| TryGetDestinationVersion | |

VelocityDB Class Library

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.BeginSession Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.BeginSession(Microsoft.Synchronization.SyncProviderPosition ,Microsoft.Synchronization.SyncSessionContext)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void BeginSession(  
    SyncProviderPosition position,  
    SyncSessionContext syncSessionContext  
)
```

VB

```
Public Overrides Sub BeginSession (  
    position As SyncProviderPosition,  
    syncSessionContext As SyncSessionContext  
)
```

C++

```
public:  
virtual void BeginSession(  
    SyncProviderPosition position,  
    SyncSessionContext^ syncSessionContext  
) override
```

F#

```
abstract BeginSession :  
    position : SyncProviderPosition *  
    syncSessionContext : SyncSessionContext -> unit  
override BeginSession :  
    position : SyncProviderPosition *  
    syncSessionContext : SyncSessionContext -> unit
```

Parameters

position

Type: **SyncProviderPosition**

[Missing <param name="position"/> documentation for

"M:VelocityDBExtensions.SyncProvider.BeginSession(Microsoft.Synchronization.SyncProviderPosition ,Microsoft.Synchronization.SyncSessionContext)"]

syncSessionContext

Type: **SyncSessionContext**

[Missing <param name="syncSessionContext"/> documentation for "M:VelocityDBExtensions.SyncProvider.BeginSession(Microsoft.Synchronization.SyncProviderPosition,Microsoft.Synchronization.SyncSessionContext)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.EndSession Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.EndSession(Microsoft.Synchronization.SyncSessionContext)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void EndSession(  
    SyncSessionContext syncSessionContext  
)
```

VB

```
Public Overrides Sub EndSession (  
    syncSessionContext As SyncSessionContext  
)
```

C++

```
public:  
virtual void EndSession(  
    SyncSessionContext^ syncSessionContext  
) override
```

F#

```
abstract EndSession :  
    syncSessionContext : SyncSessionContext -> unit  
override EndSession :  
    syncSessionContext : SyncSessionContext -> unit
```

Parameters

syncSessionContext

Type: **SyncSessionContext**

[Missing <param name="syncSessionContext"/> documentation for

"M:VelocityDBExtensions.SyncProvider.EndSession(Microsoft.Synchronization.SyncSessionContext)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.Equals Method

Determines whether the specified object is equal to the current object.

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if the specified object is equal to the current object; otherwise, `false` (`False` in Visual Basic).

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetChangeBatch Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.GetChangeBatch(System.UInt32,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override ChangeBatch GetChangeBatch(  
    uint batchSize,  
    SyncKnowledge destinationKnowledge,  
    out Object changeDataRetriever  
)
```

VB

```
Public Overrides Function GetChangeBatch (  
    batchSize As UInteger,  
    destinationKnowledge As SyncKnowledge,  
    <OutAttribute> ByRef changeDataRetriever As Object  
) As ChangeBatch
```

C++

```
public:  
virtual ChangeBatch^ GetChangeBatch(  
    unsigned int batchSize,  
    SyncKnowledge^ destinationKnowledge,  
    [OutAttribute] Object^% changeDataRetriever  
) override
```

F#

```
abstract GetChangeBatch :  
    batchSize : uint32 *  
    destinationKnowledge : SyncKnowledge *  
    changeDataRetriever : Object byref -> ChangeBatch  
override GetChangeBatch :  
    batchSize : uint32 *  
    destinationKnowledge : SyncKnowledge *  
    changeDataRetriever : Object byref -> ChangeBatch
```

Parameters

batchSize

Type: [System.UInt32](#)

[Missing <param name="batchSize"/> documentation for

"M:VelocityDBExtensions.SyncProvider.GetChangeBatch(System.UInt32,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

destinationKnowledge

Type: **SyncKnowledge**

[Missing <param name="destinationKnowledge"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetChangeBatch(System.UInt32,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

changeDataRetriever

Type: [System.Object](#)

[Missing <param name="changeDataRetriever"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetChangeBatch(System.UInt32,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

Return Value

Type: **ChangeBatch**

[Missing <returns> documentation for "M:VelocityDBExtensions.SyncProvider.GetChangeBatch(System.UInt32,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetDataRetriever Method

[Missing <summary> documentation for "M:VelocityDBExtensions.SyncProvider.GetDataRetriever"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IChangeDataRetriever GetDataRetriever ()
```

VB

```
Public Function GetDataRetriever As IChangeDataRetriever
```

C++

```
public:  
virtual IChangeDataRetriever^ GetDataRetriever () sealed
```

F#

```
abstract GetDataRetriever : unit -> IChangeDataRetriever  
override GetDataRetriever : unit -> IChangeDataRetriever
```

Return Value

Type: **IChangeDataRetriever**

[Missing <returns> documentation for "M:VelocityDBExtensions.SyncProvider.GetDataRetriever"]

Implements

[INotifyingChangeApplierTarget.GetDataRetriever\(\)](#)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetFullEnumerationChangeBatch Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override FullEnumerationChangeBatch GetFullEnumerationChangeBatch (
    uint batchSize,
    SyncId lowerEnumerationBound,
    SyncKnowledge knowledgeForDataRetrieval,
    out Object changeDataRetriever
)
```

VB

```
Public Overrides Function GetFullEnumerationChangeBatch (
    batchSize As UInteger,
    lowerEnumerationBound As SyncId,
    knowledgeForDataRetrieval As SyncKnowledge,
    <OutAttribute> ByRef changeDataRetriever As Object
) As FullEnumerationChangeBatch
```

C++

```
public:
virtual FullEnumerationChangeBatch^ GetFullEnumerationChangeBatch(
    unsigned int batchSize,
    SyncId^ lowerEnumerationBound,
    SyncKnowledge^ knowledgeForDataRetrieval,
    [OutAttribute] Object^% changeDataRetriever
) override
```

F#

```
abstract GetFullEnumerationChangeBatch :
    batchSize : uint32 *
    lowerEnumerationBound : SyncId *
    knowledgeForDataRetrieval : SyncKnowledge *
    changeDataRetriever : Object byref -> FullEnumerationChangeBatch
override GetFullEnumerationChangeBatch :
    batchSize : uint32 *
    lowerEnumerationBound : SyncId *
    knowledgeForDataRetrieval : SyncKnowledge *
    changeDataRetriever : Object byref -> FullEnumerationChangeBatch
```

Parameters

batchSize

Type: [System.UInt32](#)

[Missing <param name="batchSize"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

lowerEnumerationBound

Type: **SyncId**

[Missing <param name="lowerEnumerationBound"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

knowledgeForDataRetrieval

Type: **SyncKnowledge**

[Missing <param name="knowledgeForDataRetrieval"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

changeDataRetriever

Type: [System.Object](#)

[Missing <param name="changeDataRetriever"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

Return Value

Type: **FullEnumerationChangeBatch**

[Missing <returns> documentation for "M:VelocityDBExtensions.SyncProvider.GetFullEnumerationChangeBatch(System.UInt32,Microsoft.Synchronization.SyncId,Microsoft.Synchronization.SyncKnowledge,System.Object@)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetHashCode Method

Serves as the default hash function.

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

A hash code for the current object.

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetNextTickCount Method

[Missing <summary> documentation for "M:VelocityDBExtensions.SyncProvider.GetNextTickCount"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public ulong GetNextTickCount ()
```

VB

```
Public Function GetNextTickCount As ULong
```

C++

```
public:  
virtual unsigned long long GetNextTickCount() sealed
```

F#

```
abstract GetNextTickCount : unit -> uint64  
override GetNextTickCount : unit -> uint64
```

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.SyncProvider.GetNextTickCount"]

Implements

INotifyChangeApplierTarget.GetNextTickCount()

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.GetSyncBatchParameters Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.GetSyncBatchParameters(System.UInt32@,Microsoft.Synchronization.SyncKnowledge@)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void GetSyncBatchParameters (
    out uint batchSize,
    out SyncKnowledge knowledge
)
```

VB

```
Public Overrides Sub GetSyncBatchParameters (
    <OutAttribute> ByRef batchSize As UInteger,
    <OutAttribute> ByRef knowledge As SyncKnowledge
)
```

C++

```
public:
virtual void GetSyncBatchParameters(
    [OutAttribute] unsigned int% batchSize,
    [OutAttribute] SyncKnowledge^% knowledge
) override
```

F#

```
abstract GetSyncBatchParameters :
    batchSize : uint32 byref *
    knowledge : SyncKnowledge byref -> unit
override GetSyncBatchParameters :
    batchSize : uint32 byref *
    knowledge : SyncKnowledge byref -> unit
```

Parameters

batchSize

Type: [System.UInt32](#)

[Missing <param name="batchSize"/> documentation for

"M:VelocityDBExtensions.SyncProvider.GetSyncBatchParameters(System.UInt32@,Microsoft.Synchronization.SyncKnowledge@)"]

knowledge

Type: **SyncKnowledge**

[Missing <param name="knowledge"/> documentation for "M:VelocityDBExtensions.SyncProvider.GetSyncBatchParameters(System.UInt32@,Microsoft.Synchronization.SyncKnowledge@)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.LoadChangeData Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.LoadChangeData(Microsoft.Synchronization.LoadChangeContext)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Object LoadChangeData (
    LoadChangeContext loadChangeContext
)
```

VB

```
Public Function LoadChangeData (
    loadChangeContext As LoadChangeContext
) As Object
```

C++

```
public:
virtual Object^ LoadChangeData (
    LoadChangeContext^ loadChangeContext
) sealed
```

F#

```
abstract LoadChangeData :
    loadChangeContext : LoadChangeContext -> Object
override LoadChangeData :
    loadChangeContext : LoadChangeContext -> Object
```

Parameters

loadChangeContext

Type: **LoadChangeContext**

[Missing <param name="loadChangeContext"/> documentation for

"M:VelocityDBExtensions.SyncProvider.LoadChangeData(Microsoft.Synchronization.LoadChangeContext)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.SyncProvider.LoadChangeData(Microsoft.Synchronization.LoadChangeContext)"]

VelocityDB Class Library

Implements

IChangeDataRetriever.LoadChangeData(LoadChangeContext)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.ProcessChangeBatch Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void ProcessChangeBatch(
    ConflictResolutionPolicy resolutionPolicy,
    ChangeBatch sourceChanges,
    Object changeDataRetriever,
    SyncCallbacks syncCallbacks,
    SyncSessionStatistics sessionStatistics
)
```

VB

```
Public Overrides Sub ProcessChangeBatch (
    resolutionPolicy As ConflictResolutionPolicy,
    sourceChanges As ChangeBatch,
    changeDataRetriever As Object,
    syncCallbacks As SyncCallbacks,
    sessionStatistics As SyncSessionStatistics
)
```

C++

```
public:
virtual void ProcessChangeBatch(
    ConflictResolutionPolicy resolutionPolicy,
    ChangeBatch^ sourceChanges,
    Object^ changeDataRetriever,
    SyncCallbacks^ syncCallbacks,
    SyncSessionStatistics^ sessionStatistics
) override
```

F#

```
abstract ProcessChangeBatch :
    resolutionPolicy : ConflictResolutionPolicy *
    sourceChanges : ChangeBatch *
    changeDataRetriever : Object *
    syncCallbacks : SyncCallbacks *
    sessionStatistics : SyncSessionStatistics -> unit
override ProcessChangeBatch :
    resolutionPolicy : ConflictResolutionPolicy *
    sourceChanges : ChangeBatch *
    changeDataRetriever : Object *
```

```
syncCallbacks : SyncCallbacks *  
sessionStatistics : SyncSessionStatistics -> unit
```

Parameters

resolutionPolicy

Type: **ConflictResolutionPolicy**

[Missing <param name="resolutionPolicy"/> documentation for "M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

sourceChanges

Type: **ChangeBatch**

[Missing <param name="sourceChanges"/> documentation for "M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

changeDataRetriever

Type: [System.Object](#)

[Missing <param name="changeDataRetriever"/> documentation for "M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

syncCallbacks

Type: **SyncCallbacks**

[Missing <param name="syncCallbacks"/> documentation for "M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

sessionStatistics

Type: **SyncSessionStatistics**

[Missing <param name="sessionStatistics"/> documentation for "M:VelocityDBExtensions.SyncProvider.ProcessChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.ChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.ProcessFullEnumerationChangeBatch Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void ProcessFullEnumerationChangeBatch (
    ConflictResolutionPolicy resolutionPolicy,
    FullEnumerationChangeBatch sourceChanges,
    Object changeDataRetriever,
    SyncCallbacks syncCallbacks,
    SyncSessionStatistics sessionStatistics
)
```

VB

```
Public Overrides Sub ProcessFullEnumerationChangeBatch (
    resolutionPolicy As ConflictResolutionPolicy,
    sourceChanges As FullEnumerationChangeBatch,
    changeDataRetriever As Object,
    syncCallbacks As SyncCallbacks,
    sessionStatistics As SyncSessionStatistics
)
```

C++

```
public:
virtual void ProcessFullEnumerationChangeBatch (
    ConflictResolutionPolicy resolutionPolicy,
    FullEnumerationChangeBatch^ sourceChanges,
    Object^ changeDataRetriever,
    SyncCallbacks^ syncCallbacks,
    SyncSessionStatistics^ sessionStatistics
) override
```

F#

```
abstract ProcessFullEnumerationChangeBatch :
    resolutionPolicy : ConflictResolutionPolicy *
    sourceChanges : FullEnumerationChangeBatch *
    changeDataRetriever : Object *
    syncCallbacks : SyncCallbacks *
    sessionStatistics : SyncSessionStatistics -> unit
override ProcessFullEnumerationChangeBatch :
    resolutionPolicy : ConflictResolutionPolicy *
    sourceChanges : FullEnumerationChangeBatch *
    changeDataRetriever : Object *
```

```
syncCallbacks : SyncCallbacks *  
sessionStatistics : SyncSessionStatistics -> unit
```

Parameters

resolutionPolicy

Type: **ConflictResolutionPolicy**

[Missing <param name="resolutionPolicy"/> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

sourceChanges

Type: **FullEnumerationChangeBatch**

[Missing <param name="sourceChanges"/> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

changeDataRetriever

Type: [System.Object](#)

[Missing <param name="changeDataRetriever"/> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

syncCallbacks

Type: **SyncCallbacks**

[Missing <param name="syncCallbacks"/> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

sessionStatistics

Type: **SyncSessionStatistics**

[Missing <param name="sessionStatistics"/> documentation for

"M:VelocityDBExtensions.SyncProvider.ProcessFullEnumerationChangeBatch(Microsoft.Synchronization.ConflictResolutionPolicy,Microsoft.Synchronization.FullEnumerationChangeBatch,System.Object,Microsoft.Synchronization.SyncCallbacks,Microsoft.Synchronization.SyncSessionStatistics)"]

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.SaveChangeWithChangeUnits Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.SaveChangeWithChangeUnits(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeWithChangeUnitsContext)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void SaveChangeWithChangeUnits (
    ItemChange change,
    SaveChangeWithChangeUnitsContext context
)
```

VB

```
Public Sub SaveChangeWithChangeUnits (
    change As ItemChange,
    context As SaveChangeWithChangeUnitsContext
)
```

C++

```
public:
virtual void SaveChangeWithChangeUnits (
    ItemChange^ change,
    SaveChangeWithChangeUnitsContext^ context
) sealed
```

F#

```
abstract SaveChangeWithChangeUnits :
    change : ItemChange *
    context : SaveChangeWithChangeUnitsContext -> unit
override SaveChangeWithChangeUnits :
    change : ItemChange *
    context : SaveChangeWithChangeUnitsContext -> unit
```

Parameters

change

Type: **ItemChange**

[Missing <param name="change"/> documentation for

"M:VelocityDBExtensions.SyncProvider.SaveChangeWithChangeUnits(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeWithChangeUnitsContext)"]

context

Type: **SaveChangeWithChangeUnitsContext**

[Missing <param name="context"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveChangeWithChangeUnits(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeWithChangeUnitsContext)"]

Implements

INotifyingChangeApplierTarget.SaveChangeWithChangeUnits(ItemChange, SaveChangeWithChangeUnitsContext)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.SaveConflict Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.SaveConflict(Microsoft.Synchronization.ItemChange,System.Object,Microsoft.Synchronization.SyncKnowledge)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void SaveConflict(  
    ItemChange conflictingChange,  
    Object conflictingChangeData,  
    SyncKnowledge conflictingChangeKnowledge  
)
```

VB

```
Public Sub SaveConflict (  
    conflictingChange As ItemChange,  
    conflictingChangeData As Object,  
    conflictingChangeKnowledge As SyncKnowledge  
)
```

C++

```
public:  
virtual void SaveConflict(  
    ItemChange^ conflictingChange,  
    Object^ conflictingChangeData,  
    SyncKnowledge^ conflictingChangeKnowledge  
) sealed
```

F#

```
abstract SaveConflict :  
    conflictingChange : ItemChange *  
    conflictingChangeData : Object *  
    conflictingChangeKnowledge : SyncKnowledge -> unit  
override SaveConflict :  
    conflictingChange : ItemChange *  
    conflictingChangeData : Object *  
    conflictingChangeKnowledge : SyncKnowledge -> unit
```

Parameters

conflictingChange

Type: **ItemChange**

[Missing <param name="conflictingChange"/> documentation for

"M:VelocityDBExtensions.SyncProvider.SaveConflict(Microsoft.Synchronization.ItemChange,System.Object,Microsoft.Synchronization.SyncKnowledge)"]

conflictingChangeData

Type: [System.Object](#)

[Missing <param name="conflictingChangeData"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveConflict(Microsoft.Synchronization.ItemChange,System.Object,Microsoft.Synchronization.SyncKnowledge)"]

conflictingChangeKnowledge

Type: **SyncKnowledge**

[Missing <param name="conflictingChangeKnowledge"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveConflict(Microsoft.Synchronization.ItemChange,System.Object,Microsoft.Synchronization.SyncKnowledge)"]

Implements

INotifyingChangeApplierTarget.SaveConflict(ItemChange, Object, SyncKnowledge)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.SaveItemChange Method

[Missing <summary> documentation for "M:VelocityDBExtensions.SyncProvider.SaveItemChange(Microsoft.Synchronization.SaveChangeAction,Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeContext)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void SaveItemChange (
    SaveChangeAction saveChangeAction,
    ItemChange change,
    SaveChangeContext context
)
```

VB

```
Public Sub SaveItemChange (
    saveChangeAction As SaveChangeAction,
    change As ItemChange,
    context As SaveChangeContext
)
```

C++

```
public:
virtual void SaveItemChange (
    SaveChangeAction saveChangeAction,
    ItemChange^ change,
    SaveChangeContext^ context
) sealed
```

F#

```
abstract SaveItemChange :
    saveChangeAction : SaveChangeAction *
    change : ItemChange *
    context : SaveChangeContext -> unit
override SaveItemChange :
    saveChangeAction : SaveChangeAction *
    change : ItemChange *
    context : SaveChangeContext -> unit
```

Parameters

saveChangeAction

Type: **SaveChangeAction**

[Missing <param name="saveChangeAction"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveItemChange(Microsoft.Synchronization.SaveChangeAction,Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeContext)"]

change

Type: **ItemChange**

[Missing <param name="change"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveItemChange(Microsoft.Synchronization.SaveChangeAction,Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeContext)"]

context

Type: **SaveChangeContext**

[Missing <param name="context"/> documentation for "M:VelocityDBExtensions.SyncProvider.SaveItemChange(Microsoft.Synchronization.SaveChangeAction,Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.SaveChangeContext)"]

Implements

INotifyingChangeApplierTarget.SaveItemChange(SaveChangeAction, ItemChange, SaveChangeContext)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.StoreKnowledgeForScope Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.StoreKnowledgeForScope(Microsoft.Synchronization.SyncKnowledge,Microsoft.Synchronization.ForgottenKnowledge)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void StoreKnowledgeForScope (
    SyncKnowledge knowledge,
    ForgottenKnowledge forgottenKnowledge
)
```

VB

```
Public Sub StoreKnowledgeForScope (
    knowledge As SyncKnowledge,
    forgottenKnowledge As ForgottenKnowledge
)
```

C++

```
public:
virtual void StoreKnowledgeForScope (
    SyncKnowledge^ knowledge,
    ForgottenKnowledge^ forgottenKnowledge
) sealed
```

F#

```
abstract StoreKnowledgeForScope :
    knowledge : SyncKnowledge *
    forgottenKnowledge : ForgottenKnowledge -> unit
override StoreKnowledgeForScope :
    knowledge : SyncKnowledge *
    forgottenKnowledge : ForgottenKnowledge -> unit
```

Parameters

knowledge

Type: **SyncKnowledge**

[Missing <param name="knowledge"/> documentation for

"M:VelocityDBExtensions.SyncProvider.StoreKnowledgeForScope(Microsoft.Synchronization.SyncKnowledge,Microsoft.Synchronization.ForgottenKnowledge)"]

forgottenKnowledge

Type: **ForgottenKnowledge**

[Missing <param name="forgottenKnowledge"/> documentation for "M:VelocityDBExtensions.SyncProvider.StoreKnowledgeForScope(Microsoft.Synchronization.SyncKnowledge,Microsoft.Synchronization.ForgottenKnowledge)"]

Implements

INotifyingChangeApplierTarget.StoreKnowledgeForScope(SyncKnowledge, ForgottenKnowledge)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

SyncProvider.TryGetDestinationVersion Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.SyncProvider.TryGetDestinationVersion(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.ItemChange@)"]

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool TryGetDestinationVersion(  
    ItemChange sourceChange,  
    out ItemChange destinationVersion  
)
```

VB

```
Public Function TryGetDestinationVersion (  
    sourceChange As ItemChange,  
    <OutAttribute> ByRef destinationVersion As ItemChange  
) As Boolean
```

C++

```
public:  
virtual bool TryGetDestinationVersion(  
    ItemChange^ sourceChange,  
    [OutAttribute] ItemChange^% destinationVersion  
) sealed
```

F#

```
abstract TryGetDestinationVersion :  
    sourceChange : ItemChange *  
    destinationVersion : ItemChange byref -> bool  
override TryGetDestinationVersion :  
    sourceChange : ItemChange *  
    destinationVersion : ItemChange byref -> bool
```

Parameters

sourceChange

Type: **ItemChange**

[Missing <param name="sourceChange"/> documentation for

"M:VelocityDBExtensions.SyncProvider.TryGetDestinationVersion(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.ItemChange@)"]

destinationVersion

Type: **ItemChange**

[Missing <param name="destinationVersion"/> documentation for "M:VelocityDBExtensions.SyncProvider.TryGetDestinationVersion(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.ItemChange@)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.SyncProvider.TryGetDestinationVersion(Microsoft.Synchronization.ItemChange,Microsoft.Synchronization.ItemChange@)"]

Implements

INotifyingChangeApplierTarget.TryGetDestinationVersion(ItemChange, ItemChange)

See Also

[SyncProvider Class](#)

[VelocityDBExtensions Namespace](#)

Utilities Class

Currently only used by Database Manager

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Utilities

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class Utilities
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class Utilities
```

C++





```
[ExtensionAttribute]  
public ref class Utilities abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type Utilities = class end
```

The **Utilities** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|  | ArrayToString | Outputs a string representing an array |
|  | ToStringDetails(OptimizedPersistable, SessionBase, Boolean) | Object details as a string |
|  | ToStringDetails(OptimizedPersistable, Schema, TypeVersion, Boolean) | Currently only used by Database Manager |
|  | ToStringDetails(DataMember, Object, IOptimizedPersistable, Page, Boolean) | This is a support function for the VelocityDbBrowser. It converts a field into a string. |

VelocityDB Class Library









See Also

[VelocityDBExtensions Namespace](#)

Utilities.Utilities Methods

The [Utilities](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|   | ArrayToString | Outputs a string representing an array |
|   | ToStringDetails(OptimizedPersistable, SessionBase, Boolean) | Object details as a string |
|   | ToStringDetails(OptimizedPersistable, Schema, TypeVersion, Boolean) | Currently only used by Database Manager |
|   | ToStringDetails(DataMember, Object, IOptimizedPersistable, Page, Boolean) | This is a support function for the VelocityDbBrowser. It converts a field into a string. |

See Also

[Utilities Class](#)

[VelocityDBExtensions Namespace](#)

Utilities.ArrayToString Method

Outputs a string representing an array

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ArrayToString(  
    Array array,  
    bool isEncodedOidArray,  
    Page page,  
    Type elementType,  
    string prefix = ""  
)
```

VB

```
Public Shared Function ArrayToString (  
    array As Array,  
    isEncodedOidArray As Boolean,  
    page As Page,  
    elementType As Type,  
    Optional prefix As String = ""  
) As String
```

C++

```
public:  
static String^ ArrayToString(  
    Array^ array,  
    bool isEncodedOidArray,  
    Page^ page,  
    Type^ elementType,  
    String^ prefix = L""  
)
```

F#

```
static member ArrayToString :  
    array : Array *  
    isEncodedOidArray : bool *  
    page : Page *  
    elementType : Type *  
    ?prefix : string  
(* Defaults:  
    let _prefix = defaultArg prefix ""  
*)  
-> string
```

VelocityDB Class Library

Parameters

array

Type: [System.Array](#)

The array to represent as a string

isEncodedOidArray

Type: [System.Boolean](#)

True if [Oid](#) is encoded as a UInt32 or UInt64

page

Type: [VelocityDb.Page](#)

The page containing the array

elementType

Type: [System.Type](#)

The element [Type](#)

prefix (Optional)

Type: [System.String](#)

A prefix to use before each array element in the output string

Return Value

Type: [String](#)

A [String](#) representing the array.




See Also

[Utilities Class](#)

[VelocityDBExtensions Namespace](#)

Utilities.ToStringDetails Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|  | ToStringDetails(OptimizedPersistable, SessionBase, Boolean) | Object details as a string |
|  | ToStringDetails(OptimizedPersistable, Schema, TypeVersion, Boolean) | Currently only used by Database Manager |
|  | ToStringDetails(DataMember, Object, IOptimizedPersistable, Page, Boolean) | This is a support function for the VelocityDbBrowser. It converts a field into a string. |

See Also

[Utilities Class](#)

[VelocityDBExtensions Namespace](#)

Utilities.ToStringDetails Method (OptimizedPersistable, SessionBase, Boolean)

Object details as a string

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ToStringDetails(  
    this OptimizedPersistable pObj,  
    SessionBase session,  
    bool skipArrays = true  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function ToStringDetails (  
    pObj As OptimizedPersistable,  
    session As SessionBase,  
    Optional skipArrays As Boolean = true  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ ToStringDetails(  
    OptimizedPersistable^ pObj,  
    SessionBase^ session,  
    bool skipArrays = true  
)
```

F#

```
[<ExtensionAttribute>]  
static member ToStringDetails :  
    pObj : OptimizedPersistable *  
    session : SessionBase *  
    ?skipArrays : bool  
(* Defaults:  
    let skipArrays = defaultArg skipArrays true  
)  
-> string
```

Parameters

pObj

Type: [VelocityDb.OptimizedPersistable](#)

The object extended

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

skipArrays (Optional)

Type: [System.Boolean](#)

Indicates if string should contain detailed array data.

Return Value

Type: [String](#)

[String](#) containing all details of this object.

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [OptimizedPersistable](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[Utilities Class](#)

[ToStringDetails Overload](#)

[VelocityDBExtensions Namespace](#)

Utilities.ToStringDetails Method (OptimizedPersistable, Schema, TypeVersion, Boolean)

Currently only used by Database Manager

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ToStringDetails(  
    this OptimizedPersistable pObj,  
    Schema schema,  
    TypeVersion typeVersion,  
    bool skipArrays  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function ToStringDetails (  
    pObj As OptimizedPersistable,  
    schema As Schema,  
    typeVersion As TypeVersion,  
    skipArrays As Boolean  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ ToStringDetails(  
    OptimizedPersistable^ pObj,  
    Schema^ schema,  
    TypeVersion^ typeVersion,  
    bool skipArrays  
)
```

F#

```
[<ExtensionAttribute>]  
static member ToStringDetails :  
    pObj : OptimizedPersistable *  
    schema : Schema *  
    typeVersion : TypeVersion *  
    skipArrays : bool -> string
```

Parameters

pObj

Type: [VelocityDb.OptimizedPersistable](#)

Object for which we want detailed to string data

schema

Type: [VelocityDb.TypeInfo.Schema](#)

The active schema

typeVersion

Type: [VelocityDb.TypeInfo.TypeVersion](#)

describes the type of the pObj

skipArrays

Type: [System.Boolean](#)

if `true` include array data in generated string

Return Value

Type: [String](#)

content of an object as string

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [OptimizedPersistable](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[Utilities Class](#)

[ToStringDetails Overload](#)

[VelocityDBExtensions Namespace](#)

Utilities.ToStringDetails Method (DataMember, Object, IOptimizedPersistable, Page, Boolean)

This is a support function for the VelocityDbBrowser. It converts a field into a string.

Namespace: [VelocityDBExtensions](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string ToStringDetails(  
    DataMember member,  
    Object obj,  
    IOptimizedPersistable pObj,  
    Page page,  
    bool skipArrays  
)
```

VB

```
Public Shared Function ToStringDetails (  
    member As DataMember,  
    obj As Object,  
    pObj As IOptimizedPersistable,  
    page As Page,  
    skipArrays As Boolean  
) As String
```

C++

```
public:  
static String^ ToStringDetails(  
    DataMember^ member,  
    Object^ obj,  
    IOptimizedPersistable^ pObj,  
    Page^ page,  
    bool skipArrays  
)
```

F#

```
static member ToStringDetails :  
    member : DataMember *  
    obj : Object *  
    pObj : IOptimizedPersistable *  
    page : Page *  
    skipArrays : bool -> string
```

Parameters

member

Type: [VelocityDb.TypeInfo.DataMember](#)

VelocityDB Class Library

A field in an object

obj

Type: [System.Object](#)

The object containing the field

pObj

Type: [VelocityDb.IOptimizedPersistable](#)

[Missing <param name="pObj"/> documentation for

"M:VelocityDBExtensions.Utilities.ToStringDetails(VelocityDb.TypeInfo.DataMember,System.Object,VelocityDb.IOptimizedPersistable,VelocityDb.Page,System.Boolean)"]

page

Type: [VelocityDb.Page](#)

The page of the object

skipArrays

Type: [System.Boolean](#)

Option to skip arrays of the object

Return Value

Type: [String](#)

A [String](#) containing all details of this field.

See Also

[Utilities Class](#)




[ToStringDetails Overload](#)

[VelocityDBExtensions Namespace](#)

VelocityDBExtensions.CompressedBitArray Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions.CompressedBitArray"]

Classes

| Class | Description |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
|  EwahCompressedBitArray | |
|  EwahEnumerator | The class EwahEnumerator represents a special type of efficient enumerator iterating over (uncompressed) words of bits. |
|  RunningLengthWord | Mostly for internal use. |

EwahCompressedBitArray Class

[Missing <summary> documentation for "T:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]
public sealed class EwahCompressedBitArray : OptimizedPersistable,
    ICloneable, IEnumerable<int>, IEnumerable
```

VB

```
<SerializableAttribute>
Public NotInheritable Class EwahCompressedBitArray
    Inherits OptimizedPersistable
    Implements ICloneable, IEnumerable(Of Integer),
    IEnumerable
```

C++



```
[SerializableAttribute]
public ref class EwahCompressedBitArray sealed : public OptimizedPersistable,
    ICloneable, IEnumerable<int>, IEnumerable
```

F#



```
[<SealedAttribute>]
[<SerializableAttribute>]
type EwahCompressedBitArray =
    class
        inherit OptimizedPersistable
        interface ICloneable
        interface IEnumerable<int>
        interface IEnumerable
    end
```

The **EwahCompressedBitArray** type exposes the following members.











Constructors














| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  EwahCompressedBitArray() | Creates an empty bitmap (no bit set to true). |
|  EwahCompressedBitArray(Int32) | Sets explicitly the buffer size (in 64-bit words). The initial memory usage will be "bufferize * 64". For large poorly compressible bitmaps, using large values may improve performance. |

Properties


| Name | Description |
|---------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SizeInBits | The size in bits of the *uncompressed* bitmap represented by this compressed bitmap. Initially, the SizeInBits is zero. It is extended automatically when you set bits to true. |
|  SizeInBytes | Report the *compressed* size of the bitmap (equivalent to memory usage, after accounting for some overhead). |

Methods



| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add(Int64) | Adding words directly to the bitmap (for expert use). This is normally how you add data to the array. So you add bits in streams of 8*8 bits. |
|  Add(Int64, Int32) | Adding words directly to the bitmap (for expert use). |
|  AddStreamOfEmptyWords | For experts: You want to add many zeroes or ones? This is the method you use. |
|  And | Returns a new compressed bitmap containing the bitwise AND values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |
|  AndNot | Returns a new compressed bitmap containing the bitwise AND NOT values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |
|  bitCount | Counts the number of set (1) bits. |
|  BitmapOf | |
|  Clone | (Overrides OptimizedPersistable.Clone() .) |
|  Equals | Check to see whether the two compressed bitmaps contain the same data (effectively check whether the cardinality of a XOR is == 0. (Overrides OptimizedPersistable.Equals(Object) .) |
|  GetCardinality | reports the number of bits set to true. Running time is proportional to compressed size (as reported by SizeInBytes). |

| | |
|-------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetEnumerator | Iterator over the set bits (this is what most people will want to use to browse the content). The location of the set bits is returned, in increasing order. |
|  GetHashCode | Returns a customized hash code (based on Karp-Rabin). Naturally, if the bitmaps are equal, they will hash to the same value. (Overrides OptimizedPersistable.GetHashCode().) |
|  GetObjectData | |
|  GetPositions | get the locations of the true values as one vector. (may use more memory than GetEnumerator()) |
|  Intersects | Return true if the two EwahCompressedBitArray have both at least one true bit in the same Position. Equivalently, you could call "And" and check whether there is a set bit, but intersects will run faster if you don't need the result of the "and" operation. |
|  Not | Negate (bitwise) the current bitmap. To get a negated copy, do ((EwahCompressedBitArray) mybitmap.Clone()).not(); The running time is proportional to the compressed size (as reported by SizeInBytes). |
|  Or | Returns a new compressed bitmap containing the bitwise OR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |
|  Set | set the bit at Position i to true, the bits must be set in increasing order. For example, Set(15) and then Set(7) will fail. You must do Set(7) and then Set(15). |
|  SetSizeInBits | Change the reported size in bits of the *uncompressed* bitmap represented by this compressed bitmap. It is not possible to reduce the SizeInBits, but it can be extended. The new bits are set to false or true depending on the value of defaultvalue. |
|  Shrink | Sets the internal buffer to the minimum possible size required to contain the current bitarray. This method is useful when dealing with static bitmaps, if it is called after the final bit has been set, some memory can be free-ed. Please note, the next bit set after a call to shrink will cause the memory usage of the bit-array to double. |
|  ToDebugString | A more detailed string describing the bitmap (useful for debugging). |
|  ToString | A string describing the bitmap (Overrides OptimizedPersistable.ToString().) |
|  Xor | Returns a new compressed bitmap containing the bitwise XOR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|------------------------------|
|  | WordInBits | the number of bits in a long |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | EwahCompressedBitArray() | Creates an empty bitmap (no bit set to true). |
|  | EwahCompressedBitArray(Int32) | Sets explicitly the buffer size (in 64-bit words). The initial memory usage will be "bufferize * 64". For large poorly compressible bitmaps, using large values may improve performance. |

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray Constructor

Creates an empty bitmap (no bit set to true).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitArray ()
```

VB

```
Public Sub New
```

C++

```
public:  
EwahCompressedBitArray ()
```

F#

```
new : unit -> EwahCompressedBitArray
```

See Also

[EwahCompressedBitArray Class](#)

[EwahCompressedBitArray Overload](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray Constructor (Int32)

Sets explicitly the buffer size (in 64-bit words). The initial memory usage will be "buffersize * 64". For large poorly compressible bitmaps, using large values may improve performance.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitArray(  
    int buffersize  
)
```

VB

```
Public Sub New (  
    buffersize As Integer  
)
```

C++

```
public:  
EwahCompressedBitArray(  
    int buffersize  
)
```

F#

```
new :  
    buffersize : int -> EwahCompressedBitArray
```

Parameters

buffersize

Type: [System.Int32](#)

buffersize number of 64-bit words reserved when the object is created

See Also

[EwahCompressedBitArray Class](#)



[EwahCompressedBitArray Overload](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.EwahCompressedBitArray Properties

The [EwahCompressedBitArray](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | SizeInBits | The size in bits of the *uncompressed* bitmap represented by this compressed bitmap. Initially, the SizeInBits is zero. It is extended automatically when you set bits to true. |
|  | SizeInBytes | Report the *compressed* size of the bitmap (equivalent to memory usage, after accounting for some overhead). |

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.SizeInBits Property

The size in bits of the *uncompressed* bitmap represented by this compressed bitmap. Initially, the SizeInBits is zero. It is extended automatically when you set bits to true.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int SizeInBits { get; set; }
```

VB

```
Public Property SizeInBits As Integer  
    Get  
    Set
```

C++

```
public:  
property int SizeInBits {  
    int get ();  
    void set (int value);  
}
```

F#

```
member SizeInBits : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.SizeInBytes Property

Report the *compressed* size of the bitmap (equivalent to memory usage, after accounting for some overhead).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int SizeInBytes { get; }
```

VB

```
Public ReadOnly Property SizeInBytes As Integer  
    Get
```

C++

```
public:  
property int SizeInBytes {  
    int get ();  
}
```

F#

```
member SizeInBytes : int with get
```

Property Value

Type: [Int32](#)

See Also
















[EwahCompressedBitArray Class](#)









[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.EwahCompressedBitArray Methods



The [EwahCompressedBitArray](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add(Int64) | Adding words directly to the bitmap (for expert use). This is normally how you add data to the array. So you add bits in streams of 8*8 bits. |
|  Add(Int64, Int32) | Adding words directly to the bitmap (for expert use). |
|  AddStreamOfEmptyWords | For experts: You want to add many zeroes or ones? This is the method you use. |
|  And | Returns a new compressed bitmap containing the bitwise AND values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by <code>SizeInBytes</code>). |
|  AndNot | Returns a new compressed bitmap containing the bitwise AND NOT values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by <code>SizeInBytes</code>). |
|  bitCount | Counts the number of set (1) bits. |
|  BitmapOf | |
|  Clone | (Overrides OptimizedPersistable.Clone() .) |
|  Equals | Check to see whether the two compressed bitmaps contain the same data (effectively check whether the cardinality of a XOR is == 0. (Overrides OptimizedPersistable.Equals(Object) .) |
|  GetCardinality | reports the number of bits set to true. Running time is proportional to compressed size (as reported by <code>SizeInBytes</code>). |
|  GetEnumerator | Iterator over the set bits (this is what most people will want to use to browse the content). The location of the set bits is returned, in increasing order. |
|  GetHashCode | Returns a customized hash code (based on Karp-Rabin). Naturally, if the bitmaps are equal, they will hash to the same value. (Overrides OptimizedPersistable.GetHashCode() .) |
|  GetObjectData | |
|  GetPositions | get the locations of the true values as one vector. (may use more memory than <code>GetEnumerator()</code>) |
|  Intersects | Return true if the two <code>EwahCompressedBitArray</code> have both at least one true bit in the same <code>Position</code> . Equivalently, you could call "And" and check whether there is a set bit, but <code>intersects</code> will run faster if you don't need the result of the "and" operation. |

| | |
|-----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Not | Negate (bitwise) the current bitmap. To get a negated copy, do ((EwahCompressedBitArray) mybitmap.Clone()).not(); The running time is proportional to the compressed size (as reported by SizeInBytes). |
|  Or | Returns a new compressed bitmap containing the bitwise OR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |
|  Set | set the bit at Position i to true, the bits must be set in increasing order. For example, Set(15) and then Set(7) will fail. You must do Set(7) and then Set(15). |
|  SetSizeInBits | Change the reported size in bits of the *uncompressed* bitmap represented by this compressed bitmap. It is not possible to reduce the SizeInBits, but it can be extended. The new bits are set to false or true depending on the value of defaultvalue. |
|  Shrink | Sets the internal buffer to the minimum possible size required to contain the current bitarray. This method is useful when dealing with static bitmaps, if it is called after the final bit has been set, some memory can be free-ed. Please note, the next bit set after a call to shrink will cause the memory usage of the bit-array to double. |
|  ToDebugString | A more detailed string describing the bitmap (useful for debugging). |
|  ToString | A string describing the bitmap (Overrides OptimizedPersistable.ToString() .) |
|  Xor | Returns a new compressed bitmap containing the bitwise XOR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes). |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |



See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.Add Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add(Int64) | Adding words directly to the bitmap (for expert use). This is normally how you add data to the array. So you add bits in streams of 8*8 bits. |
|  | Add(Int64, Int32) | Adding words directly to the bitmap (for expert use). |

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitmap.Add Method (Int64)

Adding words directly to the bitmap (for expert use). This is normally how you add data to the array. So you add bits in streams of 8*8 bits.

Namespace: [VelocityDBExtensions.CompressedBitmapArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int Add(  
    long newdata  
)
```

VB

```
Public Function Add (  
    newdata As Long  
) As Integer
```

C++

```
public:  
int Add(  
    long long newdata  
)
```

F#

```
member Add :  
    newdata : int64 -> int
```

Parameters

newdata

Type: [System.Int64](#)

the word

Return Value

Type: [Int32](#)

the number of words added to the buffer

See Also

[EwahCompressedBitmapArray Class](#)

[Add Overload](#)

[VelocityDBExtensions.CompressedBitmapArray Namespace](#)

EwahCompressedBitArray.Add Method (Int64, Int32)

Adding words directly to the bitmap (for expert use).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int Add(  
    long newdata,  
    int bitsthatmatter  
)
```

VB

```
Public Function Add (  
    newdata As Long,  
    bitsthatmatter As Integer  
) As Integer
```

C++

```
public:  
int Add(  
    long long newdata,  
    int bitsthatmatter  
)
```

F#

```
member Add :  
    newdata : int64 *  
    bitsthatmatter : int -> int
```

Parameters

newdata

Type: [System.Int64](#)

the word

bitsthatmatter

Type: [System.Int32](#)

the number of significant bits (by default it should be 64)

Return Value

Type: [Int32](#)

the number of words added to the buffer

See Also

[EwahCompressedBitArray Class](#)

VelocityDB Class Library

[Add Overload](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.AddStreamOfEmptyWords Method

For experts: You want to add many zeroes or ones? This is the method you use.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int AddStreamOfEmptyWords (  
    bool v,  
    long number  
)
```

VB

```
Public Function AddStreamOfEmptyWords (  
    v As Boolean,  
    number As Long  
) As Integer
```

C++

```
public:  
int AddStreamOfEmptyWords (  
    bool v,  
    long long number  
)
```

F#

```
member AddStreamOfEmptyWords :  
    v : bool *  
    number : int64 -> int
```

Parameters

v

Type: [System.Boolean](#)

the bool value

number

Type: [System.Int64](#)

the number

Return Value

Type: [Int32](#)

the number of words added to the buffer

See Also

[EwahCompressedBitArray Class](#)

EwahCompressedBitmap.And Method

Returns a new compressed bitmap containing the bitwise AND values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes).

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitmap And(  
    EwahCompressedBitmap a  
)
```

VB

```
Public Function And (  
    a As EwahCompressedBitmap  
) As EwahCompressedBitmap
```

C++

```
public:  
EwahCompressedBitmap^ And(  
    EwahCompressedBitmap^ a  
)
```

F#

```
member And :  
    a : EwahCompressedBitmap -> EwahCompressedBitmap
```

Parameters

a

Type: [VelocityDBExtensions.CompressedBitmap.EwahCompressedBitmap](#)

the other bitmap

Return Value

Type: [EwahCompressedBitmap](#)

the EWAH compressed bitmap

See Also

[EwahCompressedBitmap Class](#)

[VelocityDBExtensions.CompressedBitmap Namespace](#)

EwahCompressedBitmap.AndNot Method

Returns a new compressed bitmap containing the bitwise AND NOT values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by `SizeInBytes`).

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitmap AndNot (  
    EwahCompressedBitmap a  
)
```

VB

```
Public Function AndNot (  
    a As EwahCompressedBitmap  
) As EwahCompressedBitmap
```

C++

```
public:  
EwahCompressedBitmap^ AndNot (  
    EwahCompressedBitmap^ a  
)
```

F#

```
member AndNot :  
    a : EwahCompressedBitmap -> EwahCompressedBitmap
```

Parameters

a

Type: [VelocityDBExtensions.CompressedBitmap.EwahCompressedBitmap](#)

the other bitmap

Return Value

Type: [EwahCompressedBitmap](#)

the EWAH compressed bitmap

See Also

[EwahCompressedBitmap Class](#)

[VelocityDBExtensions.CompressedBitmap Namespace](#)

EwahCompressedBitArray.bitCount Method

Counts the number of set (1) bits.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static ulong bitCount(  
    ulong v  
)
```

VB

```
Public Shared Function bitCount (  
    v As ULong  
) As ULong
```

C++

```
public:  
static unsigned long long bitCount(  
    unsigned long long v  
)
```

F#

```
static member bitCount :  
    v : uint64 -> uint64
```

Parameters

v

Type: [System.UInt64](#)

the value to be processed

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.bitCount(System.UInt64)"]

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.BitmapOf Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.BitmapOf(System.Int32[])"]

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static EwahCompressedBitArray BitmapOf(  
    params int[] setbits  
)
```

VB

```
Public Shared Function BitmapOf (  
    ParamArray setbits As Integer()  
) As EwahCompressedBitArray
```

C++

```
public:  
static EwahCompressedBitArray^ BitmapOf(  
    ... array<int>^ setbits  
)
```

F#

```
static member BitmapOf :  
    setbits : int[] -> EwahCompressedBitArray
```

Parameters

setbits

Type: [System.Int32\[\]](#)

[Missing <param name="setbits"/> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.BitmapOf(System.Int32[])"]

Return Value

Type: [EwahCompressedBitArray](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.BitmapOf(System.Int32[])"]

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.Clone Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.Clone"]

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Object Clone ()
```

VB

```
Public Overrides Function Clone As Object
```

C++

```
public:  
virtual Object^ Clone () override
```

F#

```
abstract Clone : unit -> Object  
override Clone : unit -> Object
```

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.Clone"]

Implements

[ICloneable.Clone\(\)](#)

[ICloneable.Clone\(\)](#)

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitmap.Equals Method

Check to see whether the two compressed bitmaps contain the same data (effectively check whether the cardinality of a XOR is == 0).

Namespace: [VelocityDBExtensions.CompressedBitmapArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool Equals(  
    Object o  
)
```

VB

```
Public Overrides Function Equals (  
    o As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ o  
) override
```

F#

```
abstract Equals :  
    o : Object -> bool  
override Equals :  
    o : Object -> bool
```

Parameters

o

Type: [System.Object](#)

the other bitmap

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitmapArray.EwahCompressedBitmapArray.Equals(System.Object)"]

See Also

[EwahCompressedBitmapArray Class](#)

[VelocityDBExtensions.CompressedBitmapArray Namespace](#)

EwahCompressedBitArray.GetCardinality Method

reports the number of bits set to true. Running time is proportional to compressed size (as reported by SizeInBytes).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public ulong GetCardinality()
```

VB

```
Public Function GetCardinality As ULong
```

C++

```
public:  
unsigned long long GetCardinality()
```

F#

```
member GetCardinality : unit -> uint64
```

Return Value

Type: [UInt64](#)

the number of bits set to true

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.GetEnumerator Method

Iterator over the set bits (this is what most people will want to use to browse the content). The location of the set bits is returned, in increasing order.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IEnumerator<int> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of Integer)
```

C++

```
public:  
virtual IEnumerator<int>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<int>  
override GetEnumerator : unit -> IEnumerator<int>
```

Return Value

Type: [IEnumerator\(Int32\)](#)

the int enumerator

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.GetHashCode Method

Returns a customized hash code (based on Karp-Rabin). Naturally, if the bitmaps are equal, they will hash to the same value.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.GetHashCode"]

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.GetObjectData Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.GetObjectData(System.Runtime.Serialization.SerializationInfo,System.Runtime.Serialization.StreamingContext)"]

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void GetObjectData(  
    SerializationInfo info,  
    StreamingContext context  
)
```

VB

```
Public Sub GetObjectData (  
    info As SerializationInfo,  
    context As StreamingContext  
)
```

C++

```
public:  
void GetObjectData(  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

F#

```
member GetObjectData :  
    info : SerializationInfo *  
    context : StreamingContext -> unit
```

Parameters

info

Type: [System.Runtime.Serialization.SerializationInfo](#)

[Missing <param name="info"/> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.GetObjectData(System.Runtime.Serialization.SerializationInfo,System.Runtime.Serialization.StreamingContext)"]

context

Type: [System.Runtime.Serialization.StreamingContext](#)

[Missing <param name="context"/> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.GetObjectData(System.Runtime.Serialization.SerializationInfo,System.Runtime.Serialization.StreamingContext)"]

VelocityDB Class Library

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.GetPositions Method

get the locations of the true values as one vector. (may use more memory than GetEnumerator())

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public List<int> GetPositions ()
```

VB

```
Public Function GetPositions As List(Of Integer)
```

C++

```
public:  
List<int>^ GetPositions ()
```

F#

```
member GetPositions : unit -> List<int>
```

Return Value

Type: [List\(Int32\)](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.GetPositions"]

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.Intersects Method

Return true if the two EwahCompressedBitArray have both at least one true bit in the same Position. Equivalently, you could call "And" and check whether there is a set bit, but intersects will run faster if you don't need the result of the "and" operation.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Intersects (
    EwahCompressedBitArray a
)
```

VB

```
Public Function Intersects (
    a As EwahCompressedBitArray
) As Boolean
```

C++

```
public:
bool Intersects (
    EwahCompressedBitArray^ a
)
```

F#

```
member Intersects :
    a : EwahCompressedBitArray -> bool
```

Parameters

a

Type: [VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray](#)

[Missing <param name="a"/> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.Intersects(VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray.Intersects(VelocityDBExtensions.CompressedBitArray.EwahCompressedBitArray)"]

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitmap.Not Method

Negate (bitwise) the current bitmap. To get a negated copy, do ((EwahCompressedBitmap) mybitmap.Clone()).not(); The running time is proportional to the compressed size (as reported by SizeInBytes).

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Not ()
```

VB

```
Public Sub Not
```

C++

```
public:  
void Not ()
```

F#

```
member Not : unit -> unit
```

See Also

[EwahCompressedBitmap Class](#)

[VelocityDBExtensions.CompressedBitmap Namespace](#)

EwahCompressedBitmap.Or Method

Returns a new compressed bitmap containing the bitwise OR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by `SizeInBytes`).

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitmap Or (  
    EwahCompressedBitmap a  
)
```

VB

```
Public Function Or (  
    a As EwahCompressedBitmap  
) As EwahCompressedBitmap
```

C++

```
public:  
EwahCompressedBitmap^ Or(  
    EwahCompressedBitmap^ a  
)
```

F#

```
member Or :  
    a : EwahCompressedBitmap -> EwahCompressedBitmap
```

Parameters

a

Type: [VelocityDBExtensions.CompressedBitmap.EwahCompressedBitmap](#)

the other bitmap

Return Value

Type: [EwahCompressedBitmap](#)

the EWAH compressed bitmap

See Also

[EwahCompressedBitmap Class](#)

[VelocityDBExtensions.CompressedBitmap Namespace](#)

EwahCompressedBitArray.Set Method

set the bit at Position *i* to true, the bits must be set in increasing order. For example, Set(15) and then Set(7) will fail. You must do Set(7) and then Set(15).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Set(  
    int i  
)
```

VB

```
Public Function Set (  
    i As Integer  
) As Boolean
```

C++

```
public:  
bool Set(  
    int i  
)
```

F#

```
member Set :  
    i : int -> bool
```

Parameters

i

Type: [System.Int32](#)

the index

Return Value

Type: [Boolean](#)

true if the value was set (always true when $i \geq \text{SizeInBits}$)

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitmap.SetSizeInBits Method

Change the reported size in bits of the *uncompressed* bitmap represented by this compressed bitmap. It is not possible to reduce the SizeInBits, but it can be extended. The new bits are set to false or true depending on the value of defaultvalue.

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool SetSizeInBits(  
    int size,  
    bool defaultvalue  
)
```

VB

```
Public Function SetSizeInBits (  
    size As Integer,  
    defaultvalue As Boolean  
) As Boolean
```

C++

```
public:  
bool SetSizeInBits(  
    int size,  
    bool defaultvalue  
)
```

F#

```
member SetSizeInBits :  
    size : int *  
    defaultvalue : bool -> bool
```

Parameters

size

Type: [System.Int32](#)

the size in bits

defaultvalue

Type: [System.Boolean](#)

the default bool value

Return Value

Type: [Boolean](#)

true if the update was possible

VelocityDB Class Library

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.Shrink Method

Sets the internal buffer to the minimum possible size required to contain the current bitarray. This method is useful when dealing with static bitmaps, if it is called after the final bit has been set, some memory can be free-ed. Please note, the next bit set after a call to shrink will cause the memory usage of the bit-array to double.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Shrink()
```

VB

```
Public Sub Shrink
```

C++

```
public:  
void Shrink()
```

F#

```
member Shrink : unit -> unit
```

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.ToDebugString Method

A more detailed string describing the bitmap (useful for debugging).

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string ToDebugString()
```

VB

```
Public Function ToDebugString As String
```

C++

```
public:  
String^ ToDebugString()
```

F#

```
member ToDebugString : unit -> string
```

Return Value

Type: [String](#)

detailed debug string

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.ToString Method

A string describing the bitmap

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

the description string

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitmap.Xor Method

Returns a new compressed bitmap containing the bitwise XOR values of the current bitmap with some other bitmap. The running time is proportional to the sum of the compressed sizes (as reported by SizeInBytes).

Namespace: [VelocityDBExtensions.CompressedBitmap](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahCompressedBitmap Xor(  
    EwahCompressedBitmap a  
)
```

VB

```
Public Function Xor (  
    a As EwahCompressedBitmap  
) As EwahCompressedBitmap
```

C++

```
public:  
EwahCompressedBitmap^ Xor(  
    EwahCompressedBitmap^ a  
)
```

F#

```
member Xor :  
    a : EwahCompressedBitmap -> EwahCompressedBitmap
```

Parameters

a

Type: [VelocityDBExtensions.CompressedBitmap.EwahCompressedBitmap](#)

the other bitmap

Return Value

Type: [EwahCompressedBitmap](#)

the EWAH compressed bitmap

See Also


[EwahCompressedBitmap Class](#)

[VelocityDBExtensions.CompressedBitmap Namespace](#)

EwahCompressedBitArray.EwahCompressedBitArray Fields

The [EwahCompressedBitArray](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|------------------------------|
|  | WordInBits | the number of bits in a long |

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahCompressedBitArray.WordInBits Field

the number of bits in a long

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const int WordInBits = 64
```

VB

```
Public Const WordInBits As Integer = 64
```

C++

```
public:  
literal int WordInBits = 64
```

F#

```
static val mutable WordInBits: int
```

Field Value

Type: [Int32](#)

See Also

[EwahCompressedBitArray Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator Class

The class EwahEnumerator represents a special type of efficient enumerator iterating over (uncompressed) words of bits.

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.CompressedBitArray.EwahEnumerator

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

```
C#
public sealed class EwahEnumerator
```


```
VB
Public NotInheritable Class EwahEnumerator
```

```
C++
public ref class EwahEnumerator sealed
```



```
F#
[<SealedAttribute>]
type EwahEnumerator = class end
```

The **EwahEnumerator** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|------------------------------------|
|  | EwahEnumerator | Instantiates a new eWAH enumerator |


Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------|
|  | Buffer | Access to the array of words |
|  | DirtyWords | Position of the dirty words represented by this running length word |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|--------------------------|
|  | HasNext | Checks for next |
|  | Next | Next running length word |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-----------------------------|
|  | _Rlw | current running length word |

See Also

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator Constructor

Instantiates a new eWAH enumerator

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public EwahEnumerator(  
    long[] a,  
    int sizeinwords  
)
```

VB

```
Public Sub New (  
    a As Long(),  
    sizeinwords As Integer  
)
```

C++

```
public:  
EwahEnumerator(  
    array<long long>^ a,  
    int sizeinwords  
)
```

F#

```
new :  
    a : int64[] *  
    sizeinwords : int -> EwahEnumerator
```

Parameters

a

Type: [System.Int64\[\]](#)

the array of words

sizeinwords

Type: [System.Int32](#)

the number of words that are significant in the array of words

See Also



[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.EwahEnumerator Properties

The [EwahEnumerator](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|---------------------------------------------------------------------|
|  | Buffer | Access to the array of words |
|  | DirtyWords | Position of the dirty words represented by this running length word |

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.Buffer Property

Access to the array of words

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long[] Buffer { get; }
```

VB

```
Public ReadOnly Property Buffer As Long()  
    Get
```

C++

```
public:  
property array<long long>^ Buffer {  
    array<long long>^ get ();  
}
```

F#

```
member Buffer : int64[] with get
```

Property Value

Type: [Int64\[\]](#)

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.DirtyWords Property

Position of the dirty words represented by this running length word

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int DirtyWords { get; }
```

VB

```
Public ReadOnly Property DirtyWords As Integer  
    Get
```

C++

```
public:  
property int DirtyWords {  
    int get ();  
}
```

F#

```
member DirtyWords : int with get
```

Property Value

Type: [Int32](#)

See Also



[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.EwahEnumerator Methods

The [EwahEnumerator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|--------------------------|
|  | HasNext | Checks for next |
|  | Next | Next running length word |

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.HasNext Method

Checks for next

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool HasNext ()
```

VB

```
Public Function HasNext As Boolean
```

C++

```
public:  
bool HasNext ()
```

F#

```
member HasNext : unit -> bool
```

Return Value

Type: [Boolean](#)

true, if successful

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.Next Method

Next running length word

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public RunningLengthWord Next ()
```

VB

```
Public Function Next As RunningLengthWord
```

C++

```
public:  
RunningLengthWord^ Next ()
```

F#

```
member Next : unit -> RunningLengthWord
```

Return Value

Type: [RunningLengthWord](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.CompressedBitArray.EwahEnumerator.Next"]

See Also


[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator.EwahEnumerator Fields

The [EwahEnumerator](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------|-----------------------------|
|  | Rlw | current running length word |

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

EwahEnumerator._Rlw Field

current running length word

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public readonly RunningLengthWord _Rlw
```

VB

```
Public ReadOnly _Rlw As RunningLengthWord
```

C++

```
public:  
initonly RunningLengthWord^ _Rlw
```

F#

```
val _Rlw: RunningLengthWord
```

Field Value

Type: [RunningLengthWord](#)

See Also

[EwahEnumerator Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord Class

Mostly for internal use.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions.CompressedBitArray.RunningLengthWord

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public sealed class RunningLengthWord : OptimizedPersistable
```

VB

```
Public NotInheritable Class RunningLengthWord  
    Inherits OptimizedPersistable
```

C++





```
public ref class RunningLengthWord sealed : public OptimizedPersistable
```

F#

```
[<SealedAttribute>]  
type RunningLengthWord =  
    class  
        inherit OptimizedPersistable  
    end
```

The **RunningLengthWord** type exposes the following members.






Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
|  Count | Return the size in uncompressed words represented by this running length word. |
|  NumberOfLiteralWords | the number of literal words |
|  RunningBit | the running bit |
|  RunningLength | the running length |



Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------------|
|  | Clone | (Overrides OptimizedPersistable.Clone() .) |
|  | ToString | (Overrides OptimizedPersistable.ToString() .) |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------|
|  | ArrayOfWords | The array of words. |
|  | LargestLiteralCount | largest number of dirty words in a run |
|  | LargestRunningLengthCount | largest number of clean words in a run |
|  | Position | The Position in array. |
|  | RunningLengthBits | number of bits dedicated to marking of the running length of clean words |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |





See Also

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.RunningLengthWord Properties

The [RunningLengthWord](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|--------------------------------------------------------------------------------|
|  | Count | Return the size in uncompressed words represented by this running length word. |
|  | NumberOfLiteralWords | the number of literal words |
|  | RunningBit | the running bit |
|  | RunningLength | the running length |

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.Count Property

Return the size in uncompressed words represented by this running length word.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long Count { get; }
```

VB

```
Public ReadOnly Property Count As Long  
    Get
```

C++

```
public:  
property long long Count {  
    long long get ();  
}
```

F#

```
member Count : int64 with get
```

Return Value

Type: [Int64](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.NumberOfLiteralWords Property

the number of literal words

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long NumberOfLiteralWords { get; set; }
```

VB

```
Public Property NumberOfLiteralWords As Long  
    Get  
    Set
```

C++

```
public:  
property long long NumberOfLiteralWords {  
    long long get ();  
    void set (long long value);  
}
```

F#

```
member NumberOfLiteralWords : int64 with get, set
```

Property Value

Type: [Int64](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.RunningBit Property

the running bit

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool RunningBit { get; set; }
```

VB

```
Public Property RunningBit As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool RunningBit {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member RunningBit : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.RunningLength Property

the running length

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long RunningLength { get; set; }
```

VB

```
Public Property RunningLength As Long  
    Get  
    Set
```

C++

```
public:  
property long long RunningLength {  
    long long get ();  
    void set (long long value);  
}
```

F#

```
member RunningLength : int64 with get, set
```

Property Value

Type: [Int64](#)

See Also



[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)



RunningLengthWord.RunningLengthWord Methods

The [RunningLengthWord](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------------|
|  | Clone | (Overrides OptimizedPersistable.Clone() .) |
|  | ToString | (Overrides OptimizedPersistable.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.Clone Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.CompressedBitArray.RunningLengthWord.Clone"]

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Object Clone()
```

VB

```
Public Overrides Function Clone As Object
```

C++

```
public:  
virtual Object^ Clone() override
```

F#

```
abstract Clone : unit -> Object  
override Clone : unit -> Object
```

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.RunningLengthWord.Clone"]

Implements

[ICloneable.Clone\(\)](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.ToString Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.CompressedBitArray.RunningLengthWord.ToString"]

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.CompressedBitArray.RunningLengthWord.ToString"]

See Also









[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.RunningLengthWord Fields

The [RunningLengthWord](#) type exposes the following members.

Fields

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------|
|  | ArrayOfWords | The array of words. |
|   | LargestLiteralCount | largest number of dirty words in a run |
|   | LargestRunningLengthCount | largest number of clean words in a run |
|  | Position | The Position in array. |
|   | RunningLengthBits | number of bits dedicated to marking of the running length of clean words |

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.ArrayOfWords Field

The array of words.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long[] ArrayOfWords
```

VB

```
Public ArrayOfWords As Long()
```

C++

```
public:  
array<long long>^ ArrayOfWords
```

F#

```
val mutable ArrayOfWords: int64[]
```

Field Value

Type: [Int64](#)[]

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.LargestLiteralCount Field

largest number of dirty words in a run

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const long LargestLiteralCount = 2147483647
```

VB

```
Public Const LargestLiteralCount As Long = 2147483647
```

C++

```
public:  
literal long long LargestLiteralCount = 2147483647
```

F#

```
static val mutable LargestLiteralCount: int64
```

Field Value

Type: [Int64](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.LargestRunningLengthCount Field

largest number of clean words in a run

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const long LargestRunningLengthCount = 4294967295
```

VB

```
Public Const LargestRunningLengthCount As Long = 4294967295
```

C++

```
public:  
literal long long LargestRunningLengthCount = 4294967295
```

F#

```
static val mutable LargestRunningLengthCount: int64
```

Field Value

Type: [Int64](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.Position Field

The Position in array.

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int Position
```

VB

```
Public Position As Integer
```

C++

```
public:  
int Position
```

F#

```
val mutable Position: int
```

Field Value

Type: [Int32](#)

See Also

[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

RunningLengthWord.RunningLengthBits Field

number of bits dedicated to marking of the running length of clean words

Namespace: [VelocityDBExtensions.CompressedBitArray](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const int RunningLengthBits = 32
```

VB

```
Public Const RunningLengthBits As Integer = 32
```

C++

```
public:  
literal int RunningLengthBits = 32
```

F#

```
static val mutable RunningLengthBits: int
```

Field Value

Type: [Int32](#)

See Also


[RunningLengthWord Class](#)

[VelocityDBExtensions.CompressedBitArray Namespace](#)

VelocityDBExtensions.Extensions.BTree Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions.Extensions.BTree"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | BTreeExtensions | A few extensions to improve performance of Linq for Objects queries We need your HELP to improve it to cover more use cases of queries! |

BTreeExtensions Class

A few extensions to improve performance of Linq for Objects queries We need your HELP to improve it to cover more use cases of queries!

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Extensions.BTree.BTreeExtensions

Namespace: [VelocityDBExtensions.Extensions.BTree](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class BTreeExtensions
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class BTreeExtensions
```

C++



```
[ExtensionAttribute]  
public ref class BTreeExtensions abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type BTreeExtensions = class end
```

The **BTreeExtensions** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------|
|  | Count(Key) | Override to improve performance over IEnumerable LINQ extension |
|  | Where(Key) | Override to improve performance over IEnumerable LINQ extension |



See Also

[VelocityDBExtensions.Extensions.BTree Namespace](#)

BTreeExtensions.BTreeExtensions Methods

The [BTreeExtensions](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------------|
|  | Count(Key) | Override to improve performance over IEnumerable LINQ extension |
|  | Where(Key) | Override to improve performance over IEnumerable LINQ extension |

See Also

[BTreeExtensions Class](#)

[VelocityDBExtensions.Extensions.BTree Namespace](#)

BTreeExtensions.Count(Key) Method

Override to improve performance over IEnumerable LINQ extension

Namespace: [VelocityDBExtensions.Extensions.BTree](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static int Count<Key>(
    this BTreeBase<Key, Key> sourceCollection
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Count(Of Key) (
    sourceCollection As BTreeBase(Of Key, Key)
) As Integer
```

C++

```
public:
    [ExtensionAttribute]
    generic<typename Key>
    static int Count(
        BTreeBase<Key, Key>^ sourceCollection
    )
```

F#

```
[<ExtensionAttribute>]
static member Count :
    sourceCollection : BTreeBase<'Key, 'Key> -> int
```

Parameters

sourceCollection

Type: [VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

the collection

Type Parameters

Key

key type

Return Value

Type: [Int32](#)

Size of the collection

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [BTreeBase](#)(**Key**, **Key**). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[BTreeExtensions Class](#)

[VelocityDBExtensions.Extensions.BTree Namespace](#)

BTreeExtensions.Where(Key) Method

Override to improve performance over IEnumerable LINQ extension

Namespace: [VelocityDBExtensions.Extensions.BTree](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static IEnumerable<Key> Where<Key>(
    this BTreeBase<Key, Key> sourceCollection,
    Expression<Func<Key, bool>> expr
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Where(Of Key) (
    sourceCollection As BTreeBase(Of Key, Key),
    expr As Expression(Of Func(Of Key, Boolean))
) As IEnumerable(Of Key)
```

C++

```
public:
[ExtensionAttribute]
generic<typename Key>
static IEnumerable<Key>^ Where(
    BTreeBase<Key, Key>^ sourceCollection,
    Expression<Func<Key, bool>>^ expr
)
```

F#

```
[<ExtensionAttribute>]
static member Where :
    sourceCollection : BTreeBase<'Key, 'Key> *
    expr : Expression<Func<'Key, bool>> -> IEnumerable<'Key>
```

Parameters

sourceCollection

Type: [VelocityDb.Collection.BTree.BTreeBase\(Key, Key\)](#)

the collection

expr

Type: [System.Linq.Expressions.Expression\(Func\(Key, Boolean\)\)](#)

an expression

Type Parameters

Key

key type

Return Value

Type: [IEnumerable\(Key\)](#)

Enumeration of collection where the expression evaluates to true

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [BTreeBase\(Key, Key\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also




[BTreeExtensions Class](#)

[VelocityDBExtensions.Extensions.BTree Namespace](#)

VelocityDBExtensions.Geo Namespace

The `VelocityDB.geohash` namespace contains classes for creating GeoHash based on Longitude and Latitude, see <https://en.wikipedia.org/wiki/Geohash> Ported to C# from Java by Mats Persson, VelocityDB, Inc. original Java code, see <https://github.com/kungfoo/geohash-java> Copyright 2010, Silvio Heuberger @ IFS www.ifs.hsr.ch This code is release under the Apache License 2.0. <http://www.apache.org/licenses/LICENSE-2.0>

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------|
|  | BoundingBox | Rectangular bounding box which is used to describe the bounds of a GeoHash query |
|  | GeoHash | See https://en.wikipedia.org/wiki/Geohash |
|  | WGS84Point | encapsulates coordinates on the earths surface. Coordinate projections might end up using this class... |

BoundingBox Class

Rectangular bounding box which is used to describe the bounds of a GeoHash query

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.BoundingBox

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]
public class BoundingBox
```

VB

```
<SerializableAttribute>
Public Class BoundingBox
```

C++




```
[SerializableAttribute]
public ref class BoundingBox
```

F#





```
[<SerializableAttribute>]
type BoundingBox = class end
```






The **BoundingBox** type exposes the following members.

Constructors







| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------|---------------------------------------------------------------|
|  | BoundingBox(BoundingBox) | Clones an existing BoundingBox |
|  | BoundingBox(WGS84Point, WGS84Point) | create a bounding box defined by two coordinates |
|  | BoundingBox(Double, Double, Double, Double) | create a bounding box defined by four Doubles |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | CenterPoint | Center WGS84Point of this BoundingBox |
|  | LatitudeSize | Max Latitude - Min Latitude |
|  | LongitudeSize | Max Longitude - Min Longitude |
|  | LowerRight | Lower right corner |

| | | |
|-----------------------------------------------------------------------------------|---------------------------|----------------------------------------------|
|  | MaxLat | Maximum Latitude of this BoundingBox |
|  | MaxLon | Maximum Longitude of this BoundingBox |
|  | MinLat | Minimum Latitude of this BoundingBox |
|  | MinLon | Minimum Longitude of this BoundingBox |
|  | UpperLeft | Upper left corner |

Methods




| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  | Contains | Is WGS84Point within BoundingBox |
|  | Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|  | ExpandToInclude | Expand this BoundingBox to include other BoundingBox |
|  | GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  | Intersects | Does this BoundingBox intersect (overlap) with other BoundingBox |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

See Also

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------|---------------------------------------------------------------|
|  | BoundingBox(BoundingBox) | Clones an existing BoundingBox |
|  | BoundingBox(WGS84Point, WGS84Point) | create a bounding box defined by two coordinates |
|  | BoundingBox(Double, Double, Double, Double) | create a bounding box defined by four Doubles |

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox Constructor (BoundingBox)

Clones an existing [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox(  
    BoundingBox that  
)
```

VB

```
Public Sub New (  
    that As BoundingBox  
)
```

C++

```
public:  
BoundingBox(  
    BoundingBox^ that  
)
```

F#

```
new :  
    that : BoundingBox -> BoundingBox
```

Parameters

that

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

to be cloned

See Also

[BoundingBox Class](#)

[BoundingBox Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox Constructor (WGS84Point, WGS84Point)

create a bounding box defined by two coordinates

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox(  
    WGS84Point p1,  
    WGS84Point p2  
)
```

VB

```
Public Sub New (  
    p1 As WGS84Point,  
    p2 As WGS84Point  
)
```

C++

```
public:  
BoundingBox(  
    WGS84Point^ p1,  
    WGS84Point^ p2  
)
```

F#

```
new :  
    p1 : WGS84Point *  
    p2 : WGS84Point -> BoundingBox
```

Parameters

p1

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="p1"/> documentation for "M:VelocityDBExtensions.Geo.BoundingBox.#ctor(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

p2

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="p2"/> documentation for "M:VelocityDBExtensions.Geo.BoundingBox.#ctor(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

See Also

[BoundingBox Class](#)

VelocityDB Class Library

[BoundingBox Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox Constructor (Double, Double, Double, Double)

create a bounding box defined by four [Doubles](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox(  
    double y1,  
    double y2,  
    double x1,  
    double x2  
)
```

VB

```
Public Sub New (  
    y1 As Double,  
    y2 As Double,  
    x1 As Double,  
    x2 As Double  
)
```

C++

```
public:  
BoundingBox(  
    double y1,  
    double y2,  
    double x1,  
    double x2  
)
```

F#

```
new :  
    y1 : float *  
    y2 : float *  
    x1 : float *  
    x2 : float -> BoundingBox
```

Parameters

y1

Type: [System.Double](#)

Y axis number 1

y2

Type: [System.Double](#)

Y axis number 2

VelocityDB Class Library

x1

Type: [System.Double](#)

X axis number 1

x2

Type: [System.Double](#)

X axis number 2

See Also

[BoundingBox Class](#)










[BoundingBox Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.BoundingBox Properties

The [BoundingBox](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------|
|  | CenterPoint | Center WGS84Point of this BoundingBox |
|  | LatitudeSize | Max Latitude - Min Latitude |
|  | LongitudeSize | Max Longitude - Min Longitude |
|  | LowerRight | Lower right corner |
|  | MaxLat | Maximum Latitude of this BoundingBox |
|  | MaxLon | Maximum Longitude of this BoundingBox |
|  | MinLat | Minimum Latitude of this BoundingBox |
|  | MinLon | Minimum Longitude of this BoundingBox |
|  | UpperLeft | Upper left corner |

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.CenterPoint Property

Center [WGS84Point](#) of this [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual WGS84Point CenterPoint { get; }
```

VB

```
Public Overridable ReadOnly Property CenterPoint As WGS84Point  
    Get
```

C++

```
public:  
virtual property WGS84Point^ CenterPoint {  
    WGS84Point^ get ();  
}
```

F#

```
abstract CenterPoint : WGS84Point with get  
override CenterPoint : WGS84Point with get
```

Property Value

Type: [WGS84Point](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.LatitudeSize Property

Max Latitude - Min Latitude

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double LatitudeSize { get; }
```

VB

```
Public Overridable ReadOnly Property LatitudeSize As Double  
    Get
```

C++

```
public:  
virtual property double LatitudeSize {  
    double get ();  
}
```

F#

```
abstract LatitudeSize : float with get  
override LatitudeSize : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.LongitudeSize Property

Max Longitude - Min Longitude

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double LongitudeSize { get; }
```

VB

```
Public Overridable ReadOnly Property LongitudeSize As Double  
    Get
```

C++

```
public:  
virtual property double LongitudeSize {  
    double get ();  
}
```

F#

```
abstract LongitudeSize : float with get  
override LongitudeSize : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.LowerRight Property

Lower right corner

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual WGS84Point LowerRight { get; }
```

VB

```
Public Overridable ReadOnly Property LowerRight As WGS84Point  
    Get
```

C++

```
public:  
virtual property WGS84Point^ LowerRight {  
    WGS84Point^ get ();  
}
```

F#

```
abstract LowerRight : WGS84Point with get  
override LowerRight : WGS84Point with get
```

Property Value

Type: [WGS84Point](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.MaxLat Property

Maximum Latitude of this [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double MaxLat { get; }
```

VB

```
Public Overridable ReadOnly Property MaxLat As Double  
    Get
```

C++

```
public:  
virtual property double MaxLat {  
    double get ();  
}
```

F#

```
abstract MaxLat : float with get  
override MaxLat : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.MaxLon Property

Maximum Longitude of this [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double MaxLon { get; }
```

VB

```
Public Overridable ReadOnly Property MaxLon As Double  
    Get
```

C++

```
public:  
virtual property double MaxLon {  
    double get ();  
}
```

F#

```
abstract MaxLon : float with get  
override MaxLon : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.MinLat Property

Minimum Latitude of this [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double MinLat { get; }
```

VB

```
Public Overridable ReadOnly Property MinLat As Double  
    Get
```

C++

```
public:  
virtual property double MinLat {  
    double get ();  
}
```

F#

```
abstract MinLat : float with get  
override MinLat : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.MinLon Property

Minimum Longitude of this [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double MinLon { get; }
```

VB

```
Public Overridable ReadOnly Property MinLon As Double  
    Get
```

C++

```
public:  
virtual property double MinLon {  
    double get ();  
}
```

F#

```
abstract MinLon : float with get  
override MinLon : float with get
```

Property Value

Type: [Double](#)

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.UpperLeft Property

Upper left corner

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual WGS84Point UpperLeft { get; }
```

VB

```
Public Overridable ReadOnly Property UpperLeft As WGS84Point  
    Get
```

C++

```
public:  
virtual property WGS84Point^ UpperLeft {  
    WGS84Point^ get ();  
}
```

F#

```
abstract UpperLeft : WGS84Point with get  
override UpperLeft : WGS84Point with get
```

Property Value

Type: [WGS84Point](#)

See Also







[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.BoundingBox Methods

The [BoundingBox](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  Contains | Is WGS84Point within BoundingBox |
|  Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|  ExpandToInclude | Expand this BoundingBox to include other BoundingBox |
|  GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  Intersects | Does this BoundingBox intersect (overlap) with other BoundingBox |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.Contains Method

Is [WGS84Point](#) within [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual bool Contains(  
    WGS84Point point  
)
```

VB

```
Public Overridable Function Contains (  
    point As WGS84Point  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    WGS84Point^ point  
)
```

F#

```
abstract Contains :  
    point : WGS84Point -> bool  
override Contains :  
    point : WGS84Point -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)
[WGS84Point](#)

for containment check

Return Value

Type: [Boolean](#)

true if contained; otherwise false

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.Equals Method

Determines whether the specified object is equal to the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if the specified object is equal to the current object; otherwise, `false` (`False` in Visual Basic).

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.ExpandToInclude Method

Expand this [BoundingBox](#) to include other [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual void ExpandToInclude (  
    BoundingBox other  
)
```

VB

```
Public Overridable Sub ExpandToInclude (  
    other As BoundingBox  
)
```

C++

```
public:  
virtual void ExpandToInclude (  
    BoundingBox^ other  
)
```

F#

```
abstract ExpandToInclude :  
    other : BoundingBox -> unit  
override ExpandToInclude :  
    other : BoundingBox -> unit
```

Parameters

other

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

Other [BoundingBox](#) to be included

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.GetHashCode Method

Serves as the default hash function.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

A hash code for the current object.

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.Intersects Method

Does this [BoundingBox](#) intersect (overlap) with other [BoundingBox](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual bool Intersects(  
    BoundingBox other  
)
```

VB

```
Public Overridable Function Intersects (  
    other As BoundingBox  
) As Boolean
```

C++

```
public:  
virtual bool Intersects(  
    BoundingBox^ other  
)
```

F#

```
abstract Intersects :  
    other : BoundingBox -> bool  
override Intersects :  
    other : BoundingBox -> bool
```

Parameters

other

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

Other [BoundingBox](#)

Return Value

Type: [Boolean](#)

true if intersect; otherwise false

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

BoundingBox.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[BoundingBox Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash Class

See <https://en.wikipedia.org/wiki/Geohash>

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.GeoHash

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]
public sealed class GeoHash : IComparable<GeoHash>
```

VB

```
<SerializableAttribute>
Public NotInheritable Class GeoHash
    Implements IComparable(Of GeoHash)
```

C++






```
[SerializableAttribute]
public ref class GeoHash sealed : IComparable<GeoHash^>
```







F#

```
[<SealedAttribute>]
[<SerializableAttribute>]
type GeoHash =
    class
        interface IComparable<GeoHash>
    end
```






















The **GeoHash** type exposes the following members.











Properties

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
|  Adjacent | returns the 8 adjacent hashes for this one. They are in the following order: N, NE, E, SE, S, SW, W, NW |
|  BoundingBox | Get BoundingBox for this GeoHash |
|  BoundingBoxCenterPoint | return the center of this s bounding box. this is rarely the same point that was used to build the hash. |
|  CharacterPrecision | Returns the number of characters that represent this hash. |
|  EasternNeighbour | Nearest GeoHash Neighbor to the East |

| | |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|  LongValue | Int64 representation of this GeoHash |
|  NorthernNeighbour | Nearest GeoHash Neighbor to the North |
|  Point | returns the <code>Point</code> that was originally used to set up this. If it was built from a base32-, this is the center point of the bounding box. |
|  SignificantBits | how many significant bits are there in this ? |
|  SouthernNeighbour | Nearest GeoHash Neighbor to the South |
|  WesternNeighbour | Nearest GeoHash Neighbor to the West |

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CompareTo | |
|  | Contains | find out if the given point lies within this hashes bounding box. <i>Note: this operation checks the bounding boxes coordinates, i.e. does not use the <code>s</code> special abilities.</i> |
|  | EnclosesCircleAroundPoint | ? |
|  | Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|   | FromBinaryString | Recreates a GeoHash from a string of 0's and 1's |
|   | FromGeohashString | build a new <code>GeoHash</code> from a base32-encoded <code>String</code> . This will also set up the hashes bounding box and other values, so it can also be used with functions like <code>within()</code> . |
|   | FromLongValue | Creates a GeoHash from a long value |
|   | FromOrd | |
|   | GeoHashStringWithCharacterPrecision | This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved. |
|  | GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  | Next() | ? |
|  | Next(Int32) | ? |
|  | Ord | ? |
|  | Prev | ? |
|   | StepsBetween | Counts the number of geohashes contained between the two (ie how many times <code>next()</code> is called to increment from |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | one to two) This value depends on the number of significant bits. |
|  | ToBase32 | get the base32 string for this . this method only makes sense, if this hash has a multiple of 5 significant bits. |
|  | ToBinaryString | Get binary String representation of this GeoHash |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |
|   | withBitPrecision | create a new with the given number of bits accuracy. This at the same time defines this hash's bounding box. |
|   | WithBitPrecision | create a new with the given number of bits accuracy. This at the same time defines this hash's bounding box. |
|   | WithCharacterPrecision | This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved. |
|  | Within | returns true if this is within the given geohash bounding box. |












See Also

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.GeoHash Properties

The [GeoHash](#) type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | Adjacent | returns the 8 adjacent hashes for this one. They are in the following order: N, NE, E, SE, S, SW, W, NW |
|  | BoundingBox | Get BoundingBox for this GeoHash |
|  | BoundingBoxCenterPoint | return the center of this s bounding box. this is rarely the same point that was used to build the hash. |
|  | CharacterPrecision | Returns the number of characters that represent this hash. |
|  | EasternNeighbour | Nearest GeoHash Neighbor to the East |
|  | LongValue | Int64 representation of this GeoHash |
|  | NorthernNeighbour | Nearest GeoHash Neighbor to the North |
|  | Point | returns the that was originally used to set up this. If it was built from a base32-, this is the center point of the bounding box. |
|  | SignificantBits | how many significant bits are there in this ? |
|  | SouthernNeighbour | Nearest GeoHash Neighbor to the South |
|  | WesternNeighbour | Nearest GeoHash Neighbor to the West |

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Adjacent Property

returns the 8 adjacent hashes for this one. They are in the following order: N, NE, E, SE, S, SW, W, NW

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash[] Adjacent { get; }
```

VB

```
Public ReadOnly Property Adjacent As GeoHash()  
    Get
```

C++

```
public:  
property array<GeoHash^>^ Adjacent {  
    array<GeoHash^>^ get ();  
}
```

F#

```
member Adjacent : GeoHash[] with get
```

Property Value

Type: [GeoHash\[\]](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.BoundingBox Property

Get **BoundingBox** for this [GeoHash](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox BoundingBox { get; }
```

VB

```
Public ReadOnly Property BoundingBox As BoundingBox  
    Get
```

C++

```
public:  
property BoundingBox^ BoundingBox {  
    BoundingBox^ get ();  
}
```

F#

```
member BoundingBox : BoundingBox with get
```

Property Value

Type: [BoundingBox](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.BoundingBoxCenterPoint Property

return the center of this s bounding box. this is rarely the same point that was used to build the hash.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public WGS84Point BoundingBoxCenterPoint { get; }
```

VB

```
Public ReadOnly Property BoundingBoxCenterPoint As WGS84Point  
    Get
```

C++

```
public:  
property WGS84Point^ BoundingBoxCenterPoint {  
    WGS84Point^ get ();  
}
```

F#

```
member BoundingBoxCenterPoint : WGS84Point with get
```

Property Value

Type: [WGS84Point](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.CharacterPrecision Property

Returns the number of characters that represent this hash.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int CharacterPrecision { get; }
```

VB

```
Public ReadOnly Property CharacterPrecision As Integer  
    Get
```

C++

```
public:  
property int CharacterPrecision {  
    int get ();  
}
```

F#

```
member CharacterPrecision : int with get
```

Property Value

Type: [Int32](#)

Exceptions

| Exception | Condition |
|-----------------------------------|--------------------------------------------------------------------------------------------|
| [! :IllegalStateException] | when the hash cannot be encoded in base32, i.e. when the precision is not a multiple of 5. |

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.EasternNeighbour Property

Nearest [GeoHash](#) Neighbor to the East

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash EasternNeighbour { get; }
```

VB

```
Public ReadOnly Property EasternNeighbour As GeoHash  
    Get
```

C++

```
public:  
property GeoHash^ EasternNeighbour {  
    GeoHash^ get ();  
}
```

F#

```
member EasternNeighbour : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.LongValue Property

[Int64](#)

representation of this [GeoHash](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

```
C#  
public long LongValue { get; }
```

```
VB  
Public ReadOnly Property LongValue As Long  
    Get
```

```
C++  
public:  
property long long LongValue {  
    long long get ();  
}
```

```
F#  
member LongValue : int64 with get
```

Property Value

Type: [Int64](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.NorthernNeighbour Property

Nearest [GeoHash](#) Neighbor to the North

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash NorthernNeighbour { get; }
```

VB

```
Public ReadOnly Property NorthernNeighbour As GeoHash  
    Get
```

C++

```
public:  
property GeoHash^ NorthernNeighbour {  
    GeoHash^ get ();  
}
```

F#

```
member NorthernNeighbour : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Point Property

returns the [GeoHash](#) that was originally used to set up this. If it was built from a base32-, this is the center point of the bounding box.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public WGS84Point Point { get; }
```

VB

```
Public ReadOnly Property Point As WGS84Point  
    Get
```

C++

```
public:  
property WGS84Point^ Point {  
    WGS84Point^ get ();  
}
```

F#

```
member Point : WGS84Point with get
```

Property Value

Type: [WGS84Point](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.WGS84Point](#)

[System.String](#)

GeoHash.SignificantBits Property

how many significant bits are there in this ?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int SignificantBits { get; }
```

VB

```
Public ReadOnly Property SignificantBits As Integer  
    Get
```

C++

```
public:  
property int SignificantBits {  
    int get ();  
}
```

F#

```
member SignificantBits : int with get
```

Property Value

Type: [Int32](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.SouthernNeighbour Property

Nearest [GeoHash](#) Neighbor to the South

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash SouthernNeighbour { get; }
```

VB

```
Public ReadOnly Property SouthernNeighbour As GeoHash  
    Get
```

C++

```
public:  
property GeoHash^ SouthernNeighbour {  
    GeoHash^ get ();  
}
```

F#

```
member SouthernNeighbour : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.WesternNeighbour Property

Nearest [GeoHash](#) Neighbor to the West

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash WesternNeighbour { get; }
```

VB

```
Public ReadOnly Property WesternNeighbour As GeoHash  
    Get
```

C++

```
public:  
property GeoHash^ WesternNeighbour {  
    GeoHash^ get ();  
}
```

F#

```
member WesternNeighbour : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also



















[GeoHash Class](#)








[VelocityDBExtensions.Geo Namespace](#)

GeoHash.GeoHash Methods

The [GeoHash](#) type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CompareTo | |
|  Contains | find out if the given point lies within this hashes bounding box. <i>Note: this operation checks the bounding boxes coordinates, i.e. does not use the s special abilities.s</i> |
|  EnclosesCircleAroundPoint | ? |
|  Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|  FromBinaryString | Recreates a GeoHash from a string of 0's and 1's |
|  FromGeohashString | build a new from a base32-encoded . This will also set up the hashes bounding box and other values, so it can also be used with functions like within(). |
|  FromLongValue | Creates a GeoHash from a long value |
|  FromOrd | |
|  GeoHashStringWithCharacterPrecision | This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved. |
|  GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  Next() | ? |
|  Next(Int32) | ? |
|  Ord | ? |
|  Prev | ? |
|  StepsBetween | Counts the number of geohashes contained between the two (ie how many times next() is called to increment from one to two) This value depends on the number of significant bits. |
|  ToBase32 | get the base32 string for this . this method only makes sense, if this hash has a multiple of 5 significant bits. |
|  ToBinaryString | Get binary String representation of this GeoHash |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | withBitPrecision | create a new <code>with</code> with the given number of bits accuracy. This at the same time defines this hash's bounding box. |
|   | WithBitPrecision | create a new <code>With</code> with the given number of bits accuracy. This at the same time defines this hash's bounding box. |
|   | WithCharacterPrecision | This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved. |
|  | Within | returns true if this is within the given geohash bounding box. |

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.CompareTo Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Geo.GeoHash.CompareTo(VelocityDBExtensions.Geo.GeoHash)"]

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int CompareTo(  
    GeoHash o  
)
```

VB

```
Public Function CompareTo (  
    o As GeoHash  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    GeoHash^ o  
) sealed
```

F#

```
abstract CompareTo :  
    o : GeoHash -> int  
override CompareTo :  
    o : GeoHash -> int
```

Parameters

o

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="o"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.CompareTo(VelocityDBExtensions.Geo.GeoHash)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.CompareTo(VelocityDBExtensions.Geo.GeoHash)"]

Implements

[IComparable\(T\).CompareTo\(T\)](#)

VelocityDB Class Library

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Contains Method

find out if the given point lies within this hashes bounding box. *Note: this operation checks the bounding boxes coordinates, i.e. does not use the special abilities.*

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    WGS84Point point  
)
```

VB

```
Public Function Contains (  
    point As WGS84Point  
) As Boolean
```

C++

```
public:  
bool Contains(  
    WGS84Point^ point  
)
```

F#

```
member Contains :  
    point : WGS84Point -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="point"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.EnclosesCircleAroundPoint Method

?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool EnclosesCircleAroundPoint(  
    WGS84Point point,  
    double radius  
)
```

VB

```
Public Function EnclosesCircleAroundPoint (  
    point As WGS84Point,  
    radius As Double  
) As Boolean
```

C++

```
public:  
bool EnclosesCircleAroundPoint(  
    WGS84Point^ point,  
    double radius  
)
```

F#

```
member EnclosesCircleAroundPoint :  
    point : WGS84Point *  
    radius : float -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

?

radius

Type: [System.Double](#)

?

Return Value

Type: [Boolean](#)

?

See Also

[GeoHash Class](#)

VelocityDB Class Library

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Equals Method

Determines whether the specified object is equal to the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if the specified object is equal to the current object; otherwise, `false` (`False` in Visual Basic).

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.FromBinaryString Method

Recreates a [GeoHash](#) from a string of 0's and 1's

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash FromBinaryString(  
    string binaryString  
)
```

VB

```
Public Shared Function FromBinaryString (  
    binaryString As String  
) As GeoHash
```

C++

```
public:  
static GeoHash^ FromBinaryString(  
    String^ binaryString  
)
```

F#

```
static member FromBinaryString :  
    binaryString : string -> GeoHash
```

Parameters

binaryString

Type: [System.String](#)

[String](#)

of 0's and 1's

Return Value

Type: [GeoHash](#)

A [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.FromGeohashString Method

build a new `GeoHash` from a base32-encoded `string`. This will also set up the hashes bounding box and other values, so it can also be used with functions like `within()`.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash FromGeohashString(  
    string geohash  
)
```

VB

```
Public Shared Function FromGeohashString (  
    geohash As String  
) As GeoHash
```

C++

```
public:  
static GeoHash^ FromGeohashString(  
    String^ geohash  
)
```

F#

```
static member FromGeohashString :  
    geohash : string -> GeoHash
```

Parameters

geohash

Type: [System.String](#)

base32-encoded [String](#)

Return Value

Type: [GeoHash](#)

A [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

[System.String](#)

GeoHash.FromLongValue Method

Creates a [GeoHash](#) from a long value

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash FromLongValue(  
    long hashVal,  
    int significantBits = 64  
)
```

VB

```
Public Shared Function FromLongValue (  
    hashVal As Long,  
    Optional significantBits As Integer = 64  
) As GeoHash
```

C++

```
public:  
static GeoHash^ FromLongValue(  
    long long hashVal,  
    int significantBits = 64  
)
```

F#

```
static member FromLongValue :  
    hashVal : int64 *  
    ?significantBits : int  
(* Defaults:  
    let _significantBits = defaultArg significantBits 64  
)  
-> GeoHash
```

Parameters

hashVal

Type: [System.Int64](#)

the [GeoHash](#) as a [Int64](#)

significantBits (Optional)

Type: [System.Int32](#)

How many bits to use from the long value (64 recommended and is default)

Return Value

Type: [GeoHash](#)

A [GeoHash](#)

VelocityDB Class Library

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.FromOrd Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.FromOrd(System.Int64,System.Int32)"]

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash FromOrd(  
    long ord,  
    int significantBits  
)
```

VB

```
Public Shared Function FromOrd (  
    ord As Long,  
    significantBits As Integer  
) As GeoHash
```

C++

```
public:  
static GeoHash^ FromOrd(  
    long long ord,  
    int significantBits  
)
```

F#

```
static member FromOrd :  
    ord : int64 *  
    significantBits : int -> GeoHash
```

Parameters

ord

Type: [System.Int64](#)

[Missing <param name="ord"/> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.FromOrd(System.Int64,System.Int32)"]

significantBits

Type: [System.Int32](#)

[Missing <param name="significantBits"/> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.FromOrd(System.Int64,System.Int32)"]

VelocityDB Class Library

Return Value

Type: [GeoHash](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.FromOrd(System.Int64,System.Int32)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.GeoHashStringWithCharacterPrecision Method

This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string GeoHashStringWithCharacterPrecision(  
    double latitude,  
    double longitude,  
    int numberOfCharacters  
)
```

VB

```
Public Shared Function GeoHashStringWithCharacterPrecision (  
    latitude As Double,  
    longitude As Double,  
    numberOfCharacters As Integer  
) As String
```

C++

```
public:  
static String^ GeoHashStringWithCharacterPrecision(  
    double latitude,  
    double longitude,  
    int numberOfCharacters  
)
```

F#

```
static member GeoHashStringWithCharacterPrecision :  
    latitude : float *  
    longitude : float *  
    numberOfCharacters : int -> string
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.GeoHashStringWithCharacterPrecision(System.Double,System.Double,System.Int32)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.GeoHashStringWithCharacterPrecision(System.Double,System.Double,System.Int32)"]

numberOfCharacters

Type: [System.Int32](#)

[Missing <param name="numberOfCharacters"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.GeoHashStringWithCharacterPrecision(System.Double,System.Double,System.Int32)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.GeoHashStringWithCharacterPrecision(System.Double,System.Double,System.Int32)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.GetHashCode Method

Serves as the default hash function.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

A hash code for the current object.



See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Next Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | Next() | ? |
|  | Next(Int32) | ? |

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Next Method

?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash Next ()
```

VB

```
Public Function Next As GeoHash
```

C++

```
public:  
GeoHash^ Next ()
```

F#

```
member Next : unit -> GeoHash
```

Return Value

Type: [GeoHash](#)

The next [GeoHash](#)

See Also

[GeoHash Class](#)

[Next Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Next Method (Int32)

?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash Next (  
    int step  
)
```

VB

```
Public Function Next (  
    step As Integer  
) As GeoHash
```

C++

```
public:  
GeoHash^ Next (  
    int step  
)
```

F#

```
member Next :  
    step : int -> GeoHash
```

Parameters

step

Type: [System.Int32](#)

?

Return Value

Type: [GeoHash](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.Next(System.Int32)"]

See Also

[GeoHash Class](#)

[Next Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Ord Method

?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public long Ord()
```

VB

```
Public Function Ord As Long
```

C++

```
public:  
long long Ord()
```

F#

```
member Ord : unit -> int64
```

Return Value

Type: [Int64](#)

?

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Prev Method

?

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHash Prev ()
```

VB

```
Public Function Prev As GeoHash
```

C++

```
public:  
GeoHash^ Prev ()
```

F#

```
member Prev : unit -> GeoHash
```

Return Value

Type: [GeoHash](#)

The previous [GeoHash](#)

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.StepsBetween Method

Counts the number of geohashes contained between the two (ie how many times next() is called to increment from one to two) This value depends on the number of significant bits.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static long StepsBetween(  
    GeoHash one,  
    GeoHash two  
)
```

VB

```
Public Shared Function StepsBetween (  
    one As GeoHash,  
    two As GeoHash  
) As Long
```

C++

```
public:  
static long long StepsBetween(  
    GeoHash^ one,  
    GeoHash^ two  
)
```

F#

```
static member StepsBetween :  
    one : GeoHash *  
    two : GeoHash -> int64
```

Parameters

one

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="one"/> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.StepsBetween(VelocityDBExtensions.Geo.GeoHash,VelocityDBExtensions.Geo.GeoHash)"]

two

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="two"/> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.StepsBetween(VelocityDBExtensions.Geo.GeoHash,VelocityDBExtensions.Geo.GeoHash)"]

VelocityDB Class Library

Return Value

Type: [Int64](#)

number of steps

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.ToBase32 Method

get the base32 string for this . this method only makes sense, if this hash has a multiple of 5 significant bits.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string ToBase32 ()
```

VB

```
Public Function ToBase32 As String
```

C++

```
public:  
String^ ToBase32 ()
```

F#

```
member ToBase32 : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.ToBase32"]

Exceptions

| Exception | Condition |
|-------------------------------------------|-------------------------------------------------------------|
| [!:IllegalStateException] | when the number of significant bits is not a multiple of 5. |

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.ToBinaryString Method

Get binary [String](#) representation of this [GeoHash](#)

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string ToBinaryString()
```

VB

```
Public Function ToBinaryString As String
```

C++

```
public:  
String^ ToBinaryString()
```

F#

```
member ToBinaryString : unit -> string
```

Return Value

Type: [String](#)

[String](#) representation

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.withBitPrecision Method

create a new `GeoHash` with the given number of bits accuracy. This at the same time defines this hash's bounding box.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash withBitPrecision(  
    WGS84Point point,  
    int numberOfBits  
)
```

VB

```
Public Shared Function withBitPrecision (  
    point As WGS84Point,  
    numberOfBits As Integer  
) As GeoHash
```

C++

```
public:  
static GeoHash^ withBitPrecision(  
    WGS84Point^ point,  
    int numberOfBits  
)
```

F#

```
static member withBitPrecision :  
    point : WGS84Point *  
    numberOfBits : int -> GeoHash
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

Location to create for

numberOfBits

Type: [System.Int32](#)

How may bits precision to use (64 is recommended)

Return Value

Type: [GeoHash](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.withBitPrecision(VelocityDBExtensions.Geo.WGS84Point,System.Int32)"]

VelocityDB Class Library

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.WithBitPrecision Method

create a new `GeoHash` with the given number of bits accuracy. This at the same time defines this hash's bounding box.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash WithBitPrecision(
    double latitude,
    double longitude,
    int numberOfBits = 64
)
```

VB

```
Public Shared Function WithBitPrecision (
    latitude As Double,
    longitude As Double,
    Optional numberOfBits As Integer = 64
) As GeoHash
```

C++

```
public:
static GeoHash^ WithBitPrecision(
    double latitude,
    double longitude,
    int numberOfBits = 64
)
```

F#

```
static member WithBitPrecision :
    latitude : float *
    longitude : float *
    ?numberOfBits : int
(* Defaults:
    let _numberOfBits = defaultArg numberOfBits 64
*)
-> GeoHash
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithBitPrecision(System.Double,System.Double,System.Int32)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithBitPrecision(System.Double,System.Double,System.Int32)"]

numberOfBits (Optional)

Type: [System.Int32](#)

[Missing <param name="numberOfBits"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithBitPrecision(System.Double,System.Double,System.Int32)"]

Return Value

Type: [GeoHash](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithBitPrecision(System.Double,System.Double,System.Int32)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

[VelocityDBExtensions.Geo.GeoHash](#)

GeoHash.WithCharacterPrecision Method

This method uses the given number of characters as the desired precision value. The hash can only be 64bits long, thus a maximum precision of 12 characters can be achieved.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static GeoHash WithCharacterPrecision(  
    double latitude,  
    double longitude,  
    int numberOfCharacters  
)
```

VB

```
Public Shared Function WithCharacterPrecision (  
    latitude As Double,  
    longitude As Double,  
    numberOfCharacters As Integer  
) As GeoHash
```

C++

```
public:  
static GeoHash^ WithCharacterPrecision(  
    double latitude,  
    double longitude,  
    int numberOfCharacters  
)
```

F#

```
static member WithCharacterPrecision :  
    latitude : float *  
    longitude : float *  
    numberOfCharacters : int -> GeoHash
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for

"M:VelocityDBExtensions.Geo.GeoHash.WithCharacterPrecision(System.Double,System.Double,System.Int32)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithCharacterPrecision(System.Double,System.Double,System.Int32)"]

numberOfCharacters

Type: [System.Int32](#)

[Missing <param name="numberOfCharacters"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithCharacterPrecision(System.Double,System.Double,System.Int32)"]

Return Value

Type: [GeoHash](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.WithCharacterPrecision(System.Double,System.Double,System.Int32)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

GeoHash.Within Method

returns true if this is within the given geohash bounding box.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Within(  
    GeoHash boundingBox  
)
```

VB

```
Public Function Within (  
    boundingBox As GeoHash  
) As Boolean
```

C++

```
public:  
bool Within(  
    GeoHash^ boundingBox  
)
```

F#

```
member Within :  
    boundingBox : GeoHash -> bool
```

Parameters

boundingBox

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="boundingBox"/> documentation for "M:VelocityDBExtensions.Geo.GeoHash.Within(VelocityDBExtensions.Geo.GeoHash)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.GeoHash.Within(VelocityDBExtensions.Geo.GeoHash)"]

See Also

[GeoHash Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point Class

encapsulates coordinates on the earths surface. Coordinate projections might end up using this class...

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.WGS84Point

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]  
public class WGS84Point
```

VB

```
<SerializableAttribute>  
Public Class WGS84Point
```

C++



```
[SerializableAttribute]  
public ref class WGS84Point
```

F#



```
[<SerializableAttribute>]  
type WGS84Point = class end
```

The **WGS84Point** type exposes the following members.




Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------|
|  | WGS84Point(WGS84Point) | Initializes a new instance of the WGS84Point class |
|  | WGS84Point(Double, Double) | Initializes a new instance of the WGS84Point class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | Latitude | |
|  | Longitude | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  | Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object).) |
|  | GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode().) |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString().) |



See Also

[VelocityDBExtensions.Geo Namespace](#)

VelocityDBExtensions.Geo.WGS84Point

WGS84Point Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|--------------------------------------------------------------------|
|  | WGS84Point(WGS84Point) | Initializes a new instance of the WGS84Point class |
|  | WGS84Point(Double, Double) | Initializes a new instance of the WGS84Point class |

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point Constructor (WGS84Point)

Initializes a new instance of the [WGS84Point](#) class

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public WGS84Point(  
    WGS84Point other  
)
```

VB

```
Public Sub New (  
    other As WGS84Point  
)
```

C++

```
public:  
WGS84Point(  
    WGS84Point^ other  
)
```

F#

```
new :  
    other : WGS84Point -> WGS84Point
```

Parameters

other

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="other"/> documentation for

"M:VelocityDBExtensions.Geo.WGS84Point.#ctor(VelocityDBExtensions.Geo.WGS84Point)"]

See Also

[WGS84Point Class](#)

[WGS84Point Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point Constructor (Double, Double)

Initializes a new instance of the [WGS84Point](#) class

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public WGS84Point(  
    double latitude,  
    double longitude  
)
```

VB

```
Public Sub New (  
    latitude As Double,  
    longitude As Double  
)
```

C++

```
public:  
WGS84Point(  
    double latitude,  
    double longitude  
)
```

F#

```
new :  
    latitude : float *  
    longitude : float -> WGS84Point
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for "M:VelocityDBExtensions.Geo.WGS84Point.#ctor(System.Double,System.Double)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for "M:VelocityDBExtensions.Geo.WGS84Point.#ctor(System.Double,System.Double)"]

See Also

[WGS84Point Class](#)



[WGS84Point Overload](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.WGS84Point Properties

The [WGS84Point](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | Latitude | |
|  | Longitude | |

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.Latitude Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.WGS84Point.Latitude"]

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double Latitude { get; }
```

VB

```
Public Overridable ReadOnly Property Latitude As Double  
    Get
```

C++

```
public:  
virtual property double Latitude {  
    double get ();  
}
```

F#

```
abstract Latitude : float with get  
override Latitude : float with get
```

Property Value

Type: [Double](#)

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.Longitude Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.WGS84Point.Longitude"]

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual double Longitude { get; }
```

VB

```
Public Overridable ReadOnly Property Longitude As Double  
    Get
```

C++

```
public:  
virtual property double Longitude {  
    double get ();  
}
```

F#

```
abstract Longitude : float with get  
override Longitude : float with get
```

Property Value

Type: [Double](#)

See Also




[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.WGS84Point Methods

The [WGS84Point](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  | Equals | Determines whether the specified object is equal to the current object. (Overrides Object.Equals(Object) .) |
|  | GetHashCode | Serves as the default hash function. (Overrides Object.GetHashCode() .) |
|  | ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.Equals Method

Determines whether the specified object is equal to the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` (`True` in Visual Basic) if the specified object is equal to the current object; otherwise, `false` (`False` in Visual Basic).

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.GetHashCode Method

Serves as the default hash function.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

A hash code for the current object.

See Also

[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

WGS84Point.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityDBExtensions.Geo](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also



[WGS84Point Class](#)

[VelocityDBExtensions.Geo Namespace](#)

VelocityDBExtensions.Geo.query Namespace

The `VelocityDB.geohash.query` namespace contains classes for querying GeoHash, see <https://en.wikipedia.org/wiki/Geohash> Ported to C# from Java by Mats Persson, VelocityDB, Inc. original Java code, see <https://github.com/kungfoo/geohash-java> Copyright 2010, Silvio Heuberger @ IFS www.ifs.hsr.ch This code is release under the Apache License 2.0. <http://www.apache.org/licenses/LICENSE-2.0>

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GeoHashBoundingBoxQuery | This class returns the hashes covering a certain bounding box. There are either 1,2 or 4 susch hashes, depending on the position of the bounding box on the geohash grid. |
|  | GeoHashCircleQuery | represents a radius search around a specific point via geohashes. Approximates the circle with a square! |

Interfaces

| | Interface | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | GeoHashQuery | |

GeoHashBoundingBoxQuery Class

This class returns the hashes covering a certain bounding box. There are either 1,2 or 4 such hashes, depending on the position of the bounding box on the geohash grid.

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]
public class GeoHashBoundingBoxQuery : GeoHashQuery
```

VB

```
<SerializableAttribute>
Public Class GeoHashBoundingBoxQuery
    Implements GeoHashQuery
```

C++


```
[SerializableAttribute]
public ref class GeoHashBoundingBoxQuery : GeoHashQuery
```

F#


```
[<SerializableAttribute>]
type GeoHashBoundingBoxQuery =
    class
        interface GeoHashQuery
    end
```

The **GeoHashBoundingBoxQuery** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------------------------|
|  | GeoHashBoundingBoxQuery | Initializes a new instance of the GeoHashBoundingBoxQuery class |




Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | BoundingBox | |
|  | SearchHashes | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------|--|
|  | WktBox | |
|-----------------------------------------------------------------------------------|------------------------|--|

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery Constructor

Initializes a new instance of the [GeoHashBoundingBoxQuery](#) class

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHashBoundingBoxQuery(  
    BoundingBox bbox  
)
```

VB

```
Public Sub New (  
    bbox As BoundingBox  
)
```

C++

```
public:  
GeoHashBoundingBoxQuery(  
    BoundingBox^ bbox  
)
```

F#

```
new :  
    bbox : BoundingBox -> GeoHashBoundingBoxQuery
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

[Missing <param name="bbox"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.#ctor(VelocityDBExtensions.Geo.BoundingBox)"]

See Also




[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.GeoHashBoundingBoxQuery Properties

The [GeoHashBoundingBoxQuery](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | BoundingBox | |
|  | SearchHashes | |
|  | WktBox | |

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.BoundingBox Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.BoundingBox"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox BoundingBox { get; }
```

VB

```
Public ReadOnly Property BoundingBox As BoundingBox  
    Get
```

C++

```
public:  
property BoundingBox^ BoundingBox {  
    BoundingBox^ get ();  
}
```

F#

```
member BoundingBox : BoundingBox with get
```

Property Value

Type: [BoundingBox](#)

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.SearchHashes Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.SearchHashes"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IList<GeoHash> SearchHashes { get; }
```

VB

```
Public ReadOnly Property SearchHashes As IList(Of GeoHash)  
    Get
```

C++

```
public:  
virtual property IList<GeoHash^>^ SearchHashes {  
    IList<GeoHash^>^ get () sealed;  
}
```

F#

```
abstract SearchHashes : IList<GeoHash> with get  
override SearchHashes : IList<GeoHash> with get
```

Property Value

Type: [IList\(GeoHash\)](#)

Implements

[GeoHashQuery.SearchHashes](#)

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.WktBox Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.WktBox"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string WktBox { get; }
```

VB

```
Public ReadOnly Property WktBox As String  
    Get
```

C++

```
public:  
virtual property String^ WktBox {  
    String^ get () sealed;  
}
```

F#

```
abstract WktBox : string with get  
override WktBox : string with get
```

Property Value

Type: [String](#)

Implements

[GeoHashQuery.WktBox](#)

See Also




[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.GeoHashBoundingBoxQuery Methods

The [GeoHashBoundingBoxQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.Contains Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.Contains Method (GeoHash)

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    GeoHash hash  
)
```

VB

```
Public Function Contains (  
    hash As GeoHash  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    GeoHash^ hash  
) sealed
```

F#

```
abstract Contains :  
    hash : GeoHash -> bool  
override Contains :  
    hash : GeoHash -> bool
```

Parameters

hash

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="hash"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

VelocityDB Class Library

Implements

[GeoHashQuery.Contains\(GeoHash\)](#)

See Also

[GeoHashBoundingBoxQuery Class](#)

[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.Contains Method (WGS84Point)

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    WGS84Point point  
)
```

VB

```
Public Function Contains (  
    point As WGS84Point  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    WGS84Point^ point  
) sealed
```

F#

```
abstract Contains :  
    point : WGS84Point -> bool  
override Contains :  
    point : WGS84Point -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="point"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

VelocityDB Class Library

Implements

[GeoHashQuery.Contains\(WGS84Point\)](#)

See Also

[GeoHashBoundingBoxQuery Class](#)

[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashBoundingBoxQuery.ToString Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.ToString"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashBoundingBoxQuery.ToString"]

See Also

[GeoHashBoundingBoxQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery Class

represents a radius search around a specific point via geohashes. Approximates the circle with a square!

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.query.GeoHashCircleQuery

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
[SerializableAttribute]
public class GeoHashCircleQuery : GeoHashQuery
```

VB

```
<SerializableAttribute>
Public Class GeoHashCircleQuery
    Implements GeoHashQuery
```

C++


```
[SerializableAttribute]
public ref class GeoHashCircleQuery : GeoHashQuery
```

F#




```
[<SerializableAttribute>]
type GeoHashCircleQuery =
    class
        interface GeoHashQuery
    end
```

The **GeoHashCircleQuery** type exposes the following members.




Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|--------------------------------------------------------------|
|  | GeoHashCircleQuery | create a with the given center point and a radius in meters. |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | BoundingBox | |
|  | SearchHashes | |
|  | WktBox | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery Constructor

create a `GeoHashCircleQuery` with the given center point and a radius in meters.

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHashCircleQuery(  
    WGS84Point center,  
    double radius  
)
```

VB

```
Public Sub New (  
    center As WGS84Point,  
    radius As Double  
)
```

C++

```
public:  
GeoHashCircleQuery(  
    WGS84Point^ center,  
    double radius  
)
```

F#

```
new :  
    center : WGS84Point *  
    radius : float -> GeoHashCircleQuery
```

Parameters

center

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="center"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.#ctor(VelocityDBExtensions.Geo.WGS84Point,System.Double)"]

radius

Type: [System.Double](#)

[Missing <param name="radius"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.#ctor(VelocityDBExtensions.Geo.WGS84Point,System.Double)"]

See Also

[GeoHashCircleQuery Class](#)

VelocityDB Class Library




[VelocityDBExtensions.Geo.query Namespace](#)

[VelocityDBExtensions.Geo.query.GeoHashCircleQuery](#)

GeoHashCircleQuery.GeoHashCircleQuery Properties

The [GeoHashCircleQuery](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | BoundingBox | |
|  | SearchHashes | |
|  | WktBox | |

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.BoundingBox Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.BoundingBox"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBox BoundingBox { get; }
```

VB

```
Public ReadOnly Property BoundingBox As BoundingBox  
    Get
```

C++

```
public:  
property BoundingBox^ BoundingBox {  
    BoundingBox^ get ();  
}
```

F#

```
member BoundingBox : BoundingBox with get
```

Property Value

Type: [BoundingBox](#)

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.SearchHashes Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.SearchHashes"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IList<GeoHash> SearchHashes { get; }
```

VB

```
Public ReadOnly Property SearchHashes As IList(Of GeoHash)  
    Get
```

C++

```
public:  
virtual property IList<GeoHash^>^ SearchHashes {  
    IList<GeoHash^>^ get () sealed;  
}
```

F#

```
abstract SearchHashes : IList<GeoHash> with get  
override SearchHashes : IList<GeoHash> with get
```

Property Value

Type: [IList\(GeoHash\)](#)

Implements

[GeoHashQuery.SearchHashes](#)

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.WktBox Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.WktBox"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string WktBox { get; }
```

VB

```
Public ReadOnly Property WktBox As String  
    Get
```

C++

```
public:  
virtual property String^ WktBox {  
    String^ get () sealed;  
}
```

F#

```
abstract WktBox : string with get  
override WktBox : string with get
```

Property Value

Type: [String](#)

Implements

[GeoHashQuery.WktBox](#)

See Also




[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.GeoHashCircleQuery Methods

The [GeoHashCircleQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.Contains Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | Contains(GeoHash) | |
|  | Contains(WGS84Point) | |

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.Contains Method (GeoHash)

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    GeoHash hash  
)
```

VB

```
Public Function Contains (  
    hash As GeoHash  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    GeoHash^ hash  
) sealed
```

F#

```
abstract Contains :  
    hash : GeoHash -> bool  
override Contains :  
    hash : GeoHash -> bool
```

Parameters

hash

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="hash"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

VelocityDB Class Library

Implements

[GeoHashQuery.Contains\(GeoHash\)](#)

See Also

[GeoHashCircleQuery Class](#)

[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.Contains Method (WGS84Point)

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    WGS84Point point  
)
```

VB

```
Public Function Contains (  
    point As WGS84Point  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    WGS84Point^ point  
) sealed
```

F#

```
abstract Contains :  
    point : WGS84Point -> bool  
override Contains :  
    point : WGS84Point -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="point"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

VelocityDB Class Library

Implements

[GeoHashQuery.Contains\(WGS84Point\)](#)

See Also

[GeoHashCircleQuery Class](#)

[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashCircleQuery.ToString Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.ToString"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.query.GeoHashCircleQuery.ToString"]

See Also

[GeoHashCircleQuery Class](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery Interface

[Missing <summary> documentation for "T:VelocityDBExtensions.Geo.query.GeoHashQuery"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public interface GeoHashQuery
```

VB

```
Public Interface GeoHashQuery
```

C++



```
public interface class GeoHashQuery
```

F#



```
type GeoHashQuery = interface end
```

The **GeoHashQuery** type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------------|
|  | SearchHashes | should return the hashes that re required to perform this search. |
|  | WktBox | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------|
|  | Contains(GeoHash) | check whether a geohash is within the hashes that make up this query. |
|  | Contains(WGS84Point) | returns whether a point lies within a query. |



See Also

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.GeoHashQuery Properties

The [GeoHashQuery](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------------|
|  | SearchHashes | should return the hashes that re required to perform this search. |
|  | WktBox | |

See Also

[GeoHashQuery Interface](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.SearchHashes Property

should return the hashes that re required to perform this search.

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
IList<GeoHash> SearchHashes { get; }
```

VB

```
ReadOnly Property SearchHashes As IList(Of GeoHash)  
    Get
```

C++

```
property IList<GeoHash^>^ SearchHashes {  
    IList<GeoHash^>^ get ();  
}
```

F#

```
abstract SearchHashes : IList<GeoHash> with get
```

Property Value

Type: [IList\(GeoHash\)](#)

See Also

[GeoHashQuery Interface](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.WktBox Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.query.GeoHashQuery.WktBox"]

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
string WktBox { get; }
```

VB

```
ReadOnly Property WktBox As String  
    Get
```

C++

```
property String^ WktBox {  
    String^ get ();  
}
```

F#

```
abstract WktBox : string with get
```

Property Value

Type: [String](#)



See Also

[GeoHashQuery Interface](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.GeoHashQuery Methods

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------|
|  | Contains(GeoHash) | check whether a geohash is within the hashes that make up this query. |
|  | Contains(WGS84Point) | returns whether a point lies within a query. |



See Also

[GeoHashQuery Interface](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.Contains Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-----------------------------------------------------------------------|
|  | Contains(GeoHash) | check whether a geohash is within the hashes that make up this query. |
|  | Contains(WGS84Point) | returns whether a point lies within a query. |

See Also

[GeoHashQuery Interface](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.Contains Method (GeoHash)

check whether a geohash is within the hashes that make up this query.

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
bool Contains(  
    GeoHash hash  
)
```

VB

```
Function Contains (  
    hash As GeoHash  
) As Boolean
```

C++

```
bool Contains(  
    GeoHash^ hash  
)
```

F#

```
abstract Contains :  
    hash : GeoHash -> bool
```

Parameters

hash

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="hash"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashQuery.Contains(VelocityDBExtensions.Geo.GeoHash)"]

See Also

[GeoHashQuery Interface](#)

[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

GeoHashQuery.Contains Method (WGS84Point)

returns whether a point lies within a query.

Namespace: [VelocityDBExtensions.Geo.query](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
bool Contains(  
    WGS84Point point  
)
```

VB

```
Function Contains (  
    point As WGS84Point  
) As Boolean
```

C++

```
bool Contains(  
    WGS84Point^ point  
)
```

F#

```
abstract Contains :  
    point : WGS84Point -> bool
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="point"/> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.query.GeoHashQuery.Contains(VelocityDBExtensions.Geo.WGS84Point)"]

See Also

[GeoHashQuery Interface](#)







[Contains Overload](#)

[VelocityDBExtensions.Geo.query Namespace](#)

VelocityDBExtensions.Geo.util Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions.Geo.util"]

Classes

| Class | Description |
|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
|  BoundingBoxGeoHashIterator | Iterate over all of the values within a bounding box at a particular resolution |
|  BoundingBoxSampler | Select random samples of geohashes within a bounding box, without replacement |
|  GeoHashSizeTable | |
|  LongUtil | |
|  TwoGeoHashBoundingBox | Ported to C# from Java by Mats Persson, VelocityDB, Inc. original Java code: Created by IntelliJ IDEA. User: kevin Date: Jan 17, 2011 Time: 12:03:47 PM |
|  VincentyGeodesy | Encapsulates Vincety's geodesy algorithm . |

BoundingBoxGeoHashIterator Class

Iterate over all of the values within a bounding box at a particular resolution

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.BoundingBoxGeoHashIterator

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class BoundingBoxGeoHashIterator : IEnumerable<GeoHash>,
    IEnumerable
```

VB

```
Public Class BoundingBoxGeoHashIterator
    Implements IEnumerable(Of GeoHash), IEnumerable
```

C++


```
public ref class BoundingBoxGeoHashIterator : IEnumerable<GeoHash^>,
    IEnumerable
```

F#

```
type BoundingBoxGeoHashIterator =
    class
        interface IEnumerable<GeoHash>
        interface IEnumerable
    end
```

The **BoundingBoxGeoHashIterator** type exposes the following members.


Constructors

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
|  BoundingBoxGeoHashIterator | Initializes a new instance of the BoundingBoxGeoHashIterator class |

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
|  BoundingBox | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | GetEnumerator | |

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxGeoHashIterator Constructor

Initializes a new instance of the [BoundingBoxGeoHashIterator](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBoxGeoHashIterator(  
    TwoGeoHashBoundingBox bbox  
)
```

VB

```
Public Sub New (  
    bbox As TwoGeoHashBoundingBox  
)
```

C++

```
public:  
BoundingBoxGeoHashIterator(  
    TwoGeoHashBoundingBox^ bbox  
)
```

F#

```
new :  
    bbox : TwoGeoHashBoundingBox -> BoundingBoxGeoHashIterator
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox](#)

[Missing <param name="bbox"/> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxGeoHashIterator.#ctor(VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox)"]

See Also


[BoundingBoxGeoHashIterator Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxGeoHashIterator.BoundingBoxGeoHashIterator Properties

The [BoundingBoxGeoHashIterator](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BoundingBox | |

See Also

[BoundingBoxGeoHashIterator Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxGeoHashIterator.BoundingBox Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.util.BoundingBoxGeoHashIterator.BoundingBox"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual TwoGeoHashBoundingBox BoundingBox { get; }
```

VB

```
Public Overridable ReadOnly Property BoundingBox As TwoGeoHashBoundingBox  
    Get
```

C++

```
public:  
virtual property TwoGeoHashBoundingBox^ BoundingBox {  
    TwoGeoHashBoundingBox^ get ();  
}
```

F#

```
abstract BoundingBox : TwoGeoHashBoundingBox with get  
override BoundingBox : TwoGeoHashBoundingBox with get
```

Property Value

Type: [TwoGeoHashBoundingBox](#)

See Also


[BoundingBoxGeoHashIterator Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxGeoHashIterator.BoundingBoxGeoHashIterator Methods

The [BoundingBoxGeoHashIterator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | GetEnumerator | |

See Also

[BoundingBoxGeoHashIterator Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxGeoHashIterator.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxGeoHashIterator.GetEnumerator"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IEnumerator<GeoHash> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of GeoHash)
```

C++

```
public:  
virtual IEnumerator<GeoHash^> GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<GeoHash>  
override GetEnumerator : unit -> IEnumerator<GeoHash>
```

Return Value

Type: [IEnumerator\(GeoHash\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxGeoHashIterator.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[BoundingBoxGeoHashIterator Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler Class

Select random samples of geohashes within a bounding box, without replacement

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.BoundingBoxSampler

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class BoundingBoxSampler
```

VB

```
Public Class BoundingBoxSampler
```

C++



```
public ref class BoundingBoxSampler
```

F#


```
type BoundingBoxSampler = class end
```

The **BoundingBoxSampler** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------|
|  | BoundingBoxSampler(TwoGeoHashBoundingBox) | Initializes a new instance of the BoundingBoxSampler class |
|  | BoundingBoxSampler(TwoGeoHashBoundingBox, Int32) | Initializes a new instance of the BoundingBoxSampler class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BoundingBox | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Next | |



VelocityDB Class Library

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------------|
|  | BoundingBoxSampler(TwoGeoHashBoundingBox) | Initializes a new instance of the BoundingBoxSampler class |
|  | BoundingBoxSampler(TwoGeoHashBoundingBox, Int32) | Initializes a new instance of the BoundingBoxSampler class |

See Also

[BoundingBoxSampler Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler Constructor (TwoGeoHashBoundingBox)

Initializes a new instance of the [BoundingBoxSampler](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBoxSampler (
    TwoGeoHashBoundingBox bbox
)
```

VB

```
Public Sub New (
    bbox As TwoGeoHashBoundingBox
)
```

C++

```
public:
    BoundingBoxSampler (
        TwoGeoHashBoundingBox^ bbox
    )
```

F#

```
new :
    bbox : TwoGeoHashBoundingBox -> BoundingBoxSampler
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox](#)

[Missing <param name="bbox"/> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxSampler.#ctor(VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox)"]

Exceptions

| Exception | Condition |
|-------------------------------------|------------------------------------------------------------------------------------|
| [!:IllegalArgumentException] | if the number of geohashes contained in the bounding box exceeds Integer.MAX_VALUE |

See Also

[BoundingBoxSampler Class](#)

[BoundingBoxSampler Overload](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler Constructor (TwoGeoHashBoundingBox, Int32)

Initializes a new instance of the [BoundingBoxSampler](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BoundingBoxSampler(  
    TwoGeoHashBoundingBox bbox,  
    int seed  
)
```

VB

```
Public Sub New (  
    bbox As TwoGeoHashBoundingBox,  
    seed As Integer  
)
```

C++

```
public:  
BoundingBoxSampler(  
    TwoGeoHashBoundingBox^ bbox,  
    int seed  
)
```

F#

```
new :  
    bbox : TwoGeoHashBoundingBox *  
    seed : int -> BoundingBoxSampler
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox](#)

[Missing <param name="bbox"/> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxSampler.#ctor(VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox,System.Int32)"]

seed

Type: [System.Int32](#)

[Missing <param name="seed"/> documentation for

"M:VelocityDBExtensions.Geo.util.BoundingBoxSampler.#ctor(VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox,System.Int32)"]

See Also

[BoundingBoxSampler Class](#)

VelocityDB Class Library


[BoundingBoxSampler Overload](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler.BoundingBoxSampler Properties

The [BoundingBoxSampler](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BoundingBox | |

See Also

[BoundingBoxSampler Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler.BoundingBox Property

[Missing <summary> documentation for "P:VelocityDBExtensions.Geo.util.BoundingBoxSampler.BoundingBox"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual TwoGeoHashBoundingBox BoundingBox { get; }
```

VB

```
Public Overridable ReadOnly Property BoundingBox As TwoGeoHashBoundingBox  
    Get
```

C++

```
public:  
virtual property TwoGeoHashBoundingBox^ BoundingBox {  
    TwoGeoHashBoundingBox^ get ();  
}
```

F#

```
abstract BoundingBox : TwoGeoHashBoundingBox with get  
override BoundingBox : TwoGeoHashBoundingBox with get
```

Property Value

Type: [TwoGeoHashBoundingBox](#)

See Also


[BoundingBoxSampler Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler.BoundingBoxSampler Methods

The [BoundingBoxSampler](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Next | |

See Also

[BoundingBoxSampler Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

BoundingBoxSampler.Next Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Geo.util.BoundingBoxSampler.Next"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual GeoHash Next ()
```

VB

```
Public Overridable Function Next As GeoHash
```

C++

```
public:  
virtual GeoHash^ Next ()
```

F#

```
abstract Next : unit -> GeoHash  
override Next : unit -> GeoHash
```

Return Value

Type: [GeoHash](#)

next sample, or NULL if all samples have been returned

See Also

[BoundingBoxSampler Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

GeoHashSizeTable Class

[Missing <summary> documentation for "T:VelocityDBExtensions.Geo.util.GeoHashSizeTable"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.GeoHashSizeTable

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#
`public class GeoHashSizeTable`


VB
`Public Class GeoHashSizeTable`

C++
`public ref class GeoHashSizeTable`


F#
`type GeoHashSizeTable = class end`

The **GeoHashSizeTable** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
|  | GeoHashSizeTable | Initializes a new instance of the GeoHashSizeTable class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------|-------------|
|  | NumberOfBitsForOverlappingGeoHash | |

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

GeoHashSizeTable Constructor

Initializes a new instance of the [GeoHashSizeTable](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public GeoHashSizeTable ()
```

VB

```
Public Sub New
```

C++

```
public:  
GeoHashSizeTable ()
```

F#

```
new : unit -> GeoHashSizeTable
```

See Also


[GeoHashSizeTable Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

GeoHashSizeTable.GeoHashSizeTable Methods

The [GeoHashSizeTable](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------|-------------|
|  | NumberOfBitsForOverlappingGeoHash | |

See Also

[GeoHashSizeTable Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

GeoHashSizeTable.NumberOfBitsForOverlappingGeoHash Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.GeoHashSizeTable.NumberOfBitsForOverlappingGeoHash(VelocityDBExtensions.Geo.BoundingBox)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static int NumberOfBitsForOverlappingGeoHash(  
    BoundingBox boundingBox  
)
```

VB

```
Public Shared Function NumberOfBitsForOverlappingGeoHash (  
    boundingBox As BoundingBox  
) As Integer
```

C++

```
public:  
static int NumberOfBitsForOverlappingGeoHash(  
    BoundingBox^ boundingBox  
)
```

F#

```
static member NumberOfBitsForOverlappingGeoHash :  
    boundingBox : BoundingBox -> int
```

Parameters

boundingBox

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

[Missing <param name="boundingBox"/> documentation for

"M:VelocityDBExtensions.Geo.util.GeoHashSizeTable.NumberOfBitsForOverlappingGeoHash(VelocityDBExtensions.Geo.BoundingBox)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.GeoHashSizeTable.NumberOfBitsForOverlappingGeoHash(VelocityDBExtensions.Geo.BoundingBox)"]

See Also

[GeoHashSizeTable Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil Class

[Missing <summary> documentation for "T:VelocityDBExtensions.Geo.util.LongUtil"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.LongUtil

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

```
C#  
public class LongUtil
```


```
VB  
Public Class LongUtil
```

```
C++  
public ref class LongUtil
```


```
F#  
type LongUtil = class end
```

The **LongUtil** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------|
|  | LongUtil | Initializes a new instance of the LongUtil class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------|
|  | CommonPrefixLength | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FIRST_BIT | |

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil Constructor

Initializes a new instance of the [LongUtil](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public LongUtil ()
```

VB

```
Public Sub New
```

C++

```
public:  
LongUtil ()
```

F#

```
new : unit -> LongUtil
```

See Also

[LongUtil Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil.LongUtil Methods

The [LongUtil](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------|
|  | CommonPrefixLength | |

See Also

[LongUtil Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil.CommonPrefixLength Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.LongUtil.CommonPrefixLength(System.Int64,System.Int64)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static int CommonPrefixLength(  
    long a,  
    long b  
)
```

VB

```
Public Shared Function CommonPrefixLength (  
    a As Long,  
    b As Long  
) As Integer
```

C++

```
public:  
static int CommonPrefixLength(  
    long long a,  
    long long b  
)
```

F#

```
static member CommonPrefixLength :  
    a : int64 *  
    b : int64 -> int
```

Parameters

a

Type: [System.Int64](#)

[Missing <param name="a"/> documentation for

"M:VelocityDBExtensions.Geo.util.LongUtil.CommonPrefixLength(System.Int64,System.Int64)"]

b

Type: [System.Int64](#)

[Missing <param name="b"/> documentation for

"M:VelocityDBExtensions.Geo.util.LongUtil.CommonPrefixLength(System.Int64,System.Int64)"]

VelocityDB Class Library

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.LongUtil.CommonPrefixLength(System.Int64,System.Int64)"]

See Also


[LongUtil Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil.LongUtil Fields

The [LongUtil](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FIRST_BIT | |

See Also

[LongUtil Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

LongUtil.FIRST_BIT Field

[Missing <summary> documentation for "F:VelocityDBExtensions.Geo.util.LongUtil.FIRST_BIT"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const long FIRST_BIT = -9223372036854775808
```

VB

```
Public Const FIRST_BIT As Long = -9223372036854775808
```

C++

```
public:  
literal long long FIRST_BIT = -9223372036854775808
```

F#

```
static val mutable FIRST_BIT: int64
```

Field Value

Type: [Int64](#)

See Also

[LongUtil Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox Class

Ported to C# from Java by Mats Persson, VelocityDB, Inc. original Java code: Created by IntelliJ IDEA.

User: kevin Date: Jan 17, 2011 Time: 12:03:47 PM

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class TwoGeoHashBoundingBox
```

VB

```
Public Class TwoGeoHashBoundingBox
```

C++


```
public ref class TwoGeoHashBoundingBox
```

F#




```
type TwoGeoHashBoundingBox = class end
```

The **TwoGeoHashBoundingBox** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------|
|  | TwoGeoHashBoundingBox | Initializes a new instance of the TwoGeoHashBoundingBox class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BottomLeft | |
|  | BoundingBox | |
|  | TopRight | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-------------|
|  | FromBase32 | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|----------------------------------------|--|
|  | ToBase32 | |
|  | WithBitPrecision | |
|  | WithCharacterPrecision | |

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox Constructor

Initializes a new instance of the [TwoGeoHashBoundingBox](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public TwoGeoHashBoundingBox(  
    GeoHash bottomLeft,  
    GeoHash topRight  
)
```

VB

```
Public Sub New (  
    bottomLeft As GeoHash,  
    topRight As GeoHash  
)
```

C++

```
public:  
TwoGeoHashBoundingBox(  
    GeoHash^ bottomLeft,  
    GeoHash^ topRight  
)
```

F#

```
new :  
    bottomLeft : GeoHash *  
    topRight : GeoHash -> TwoGeoHashBoundingBox
```

Parameters

bottomLeft

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="bottomLeft"/> documentation for "M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.#ctor(VelocityDBExtensions.Geo.GeoHash,VelocityDBExtensions.Geo.GeoHash)"]

topRight

Type: [VelocityDBExtensions.Geo.GeoHash](#)

[Missing <param name="topRight"/> documentation for "M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.#ctor(VelocityDBExtensions.Geo.GeoHash,VelocityDBExtensions.Geo.GeoHash)"]




See Also

[TwoGeoHashBoundingBox Class](#)

TwoGeoHashBoundingBox.TwoGeoHashBoundingBox Properties

The [TwoGeoHashBoundingBox](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BottomLeft | |
|  | BoundingBox | |
|  | TopRight | |

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.BottomLeft Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.BottomLeft"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual GeoHash BottomLeft { get; }
```

VB

```
Public Overridable ReadOnly Property BottomLeft As GeoHash  
    Get
```

C++

```
public:  
virtual property GeoHash^ BottomLeft {  
    GeoHash^ get ();  
}
```

F#

```
abstract BottomLeft : GeoHash with get  
override BottomLeft : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.BoundingBox Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.BoundingBox"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual BoundingBox BoundingBox { get; }
```

VB

```
Public Overridable ReadOnly Property BoundingBox As BoundingBox  
    Get
```

C++

```
public:  
virtual property BoundingBox^ BoundingBox {  
    BoundingBox^ get ();  
}
```

F#

```
abstract BoundingBox : BoundingBox with get  
override BoundingBox : BoundingBox with get
```

Property Value

Type: [BoundingBox](#)

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.TopRight Property

[Missing <summary> documentation for

"P:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.TopRight"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual GeoHash TopRight { get; }
```

VB

```
Public Overridable ReadOnly Property TopRight As GeoHash  
    Get
```

C++

```
public:  
virtual property GeoHash^ TopRight {  
    GeoHash^ get ();  
}
```

F#

```
abstract TopRight : GeoHash with get  
override TopRight : GeoHash with get
```

Property Value

Type: [GeoHash](#)

See Also





[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.TwoGeoHashBoundingBox Methods

The [TwoGeoHashBoundingBox](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|-------------|
|  | FromBase32 | |
|  | ToBase32 | |
|  | WithBitPrecision | |
|  | WithCharacterPrecision | |

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.FromBase32 Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.FromBase32(System.String)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static TwoGeoHashBoundingBox FromBase32(  
    string base32  
)
```

VB

```
Public Shared Function FromBase32 (  
    base32 As String  
) As TwoGeoHashBoundingBox
```

C++

```
public:  
static TwoGeoHashBoundingBox^ FromBase32(  
    String^ base32  
)
```

F#

```
static member FromBase32 :  
    base32 : string -> TwoGeoHashBoundingBox
```

Parameters

base32

Type: [System.String](#)

[Missing <param name="base32"/> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.FromBase32(System.String)"]

Return Value

Type: [TwoGeoHashBoundingBox](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.FromBase32(System.String)"]

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.ToBase32 Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.ToBase32"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual string ToBase32 ()
```

VB

```
Public Overridable Function ToBase32 As String
```

C++

```
public:  
virtual String^ ToBase32 ()
```

F#

```
abstract ToBase32 : unit -> string  
override ToBase32 : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.ToBase32"]

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.WithBitPrecision Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithBitPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static TwoGeoHashBoundingBox WithBitPrecision(  
    BoundingBox bbox,  
    int numberOfBits  
)
```

VB

```
Public Shared Function WithBitPrecision (  
    bbox As BoundingBox,  
    numberOfBits As Integer  
) As TwoGeoHashBoundingBox
```

C++

```
public:  
static TwoGeoHashBoundingBox^ WithBitPrecision(  
    BoundingBox^ bbox,  
    int numberOfBits  
)
```

F#

```
static member WithBitPrecision :  
    bbox : BoundingBox *  
    numberOfBits : int -> TwoGeoHashBoundingBox
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

[Missing <param name="bbox"/> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithBitPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

numberOfBits

Type: [System.Int32](#)

[Missing <param name="numberOfBits"/> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithBitPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

VelocityDB Class Library

Return Value

Type: [TwoGeoHashBoundingBox](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithBitPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

TwoGeoHashBoundingBox.WithCharacterPrecision Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithCharacterPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static TwoGeoHashBoundingBox WithCharacterPrecision (
    BoundingBox bbox,
    int numberOfCharacters
)
```

VB

```
Public Shared Function WithCharacterPrecision (
    bbox As BoundingBox,
    numberOfCharacters As Integer
) As TwoGeoHashBoundingBox
```

C++

```
public:
static TwoGeoHashBoundingBox^ WithCharacterPrecision(
    BoundingBox^ bbox,
    int numberOfCharacters
)
```

F#

```
static member WithCharacterPrecision :
    bbox : BoundingBox *
    numberOfCharacters : int -> TwoGeoHashBoundingBox
```

Parameters

bbox

Type: [VelocityDBExtensions.Geo.BoundingBox](#)

[Missing <param name="bbox"/> documentation for "M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithCharacterPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

numberOfCharacters

Type: [System.Int32](#)

[Missing <param name="numberOfCharacters"/> documentation for "M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithCharacterPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

VelocityDB Class Library

Return Value

Type: [TwoGeoHashBoundingBox](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.TwoGeoHashBoundingBox.WithCharacterPrecision(VelocityDBExtensions.Geo.BoundingBox,System.Int32)"]

See Also

[TwoGeoHashBoundingBox Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy Class

Encapsulates Vincety's geodesy algorithm .

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Geo.util.VincentyGeodesy

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class VincentyGeodesy
```

VB

```
Public Class VincentyGeodesy
```

C++


```
public ref class VincentyGeodesy
```

F#

```
type VincentyGeodesy = class end
```

The **VincentyGeodesy** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|----------------------------------------------------------------|
|  | VincentyGeodesy | Initializes a new instance of the VincentyGeodesy class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | DistanceInMeters | |
|  | MoveInDirection | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | DegToRad | |
|  | EPSILON | |

VelocityDB Class Library

See Also

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy Constructor

Initializes a new instance of the [VincentyGeodesy](#) class

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public VincentyGeodesy ()
```

VB

```
Public Sub New
```

C++

```
public:  
VincentyGeodesy ()
```

F#

```
new : unit -> VincentyGeodesy
```

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.VincentyGeodesy Methods

The [VincentyGeodesy](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-------------|
|  | DistanceInMeters | |
|  | MoveInDirection | |

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.DistanceInMeters Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.DistanceInMeters(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static double DistanceInMeters (
    WGS84Point foo,
    WGS84Point bar
)
```

VB

```
Public Shared Function DistanceInMeters (
    foo As WGS84Point,
    bar As WGS84Point
) As Double
```

C++

```
public:
static double DistanceInMeters(
    WGS84Point^ foo,
    WGS84Point^ bar
)
```

F#

```
static member DistanceInMeters :
    foo : WGS84Point *
    bar : WGS84Point -> float
```

Parameters

foo

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="foo"/> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.DistanceInMeters(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

bar

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="bar"/> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.DistanceInMeters(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

VelocityDB Class Library

Return Value

Type: [Double](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.DistanceInMeters(VelocityDBExtensions.Geo.WGS84Point,VelocityDBExtensions.Geo.WGS84Point)"]

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.MoveInDirection Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.MoveInDirection(VelocityDBExtensions.Geo.WGS84Point,System.Double,System.Double)"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static WGS84Point MoveInDirection(  
    WGS84Point point,  
    double bearingInDegrees,  
    double distanceInMeters  
)
```

VB

```
Public Shared Function MoveInDirection (  
    point As WGS84Point,  
    bearingInDegrees As Double,  
    distanceInMeters As Double  
) As WGS84Point
```

C++

```
public:  
static WGS84Point^ MoveInDirection(  
    WGS84Point^ point,  
    double bearingInDegrees,  
    double distanceInMeters  
)
```

F#

```
static member MoveInDirection :  
    point : WGS84Point *  
    bearingInDegrees : float *  
    distanceInMeters : float -> WGS84Point
```

Parameters

point

Type: [VelocityDBExtensions.Geo.WGS84Point](#)

[Missing <param name="point"/> documentation for

"M:VelocityDBExtensions.Geo.util.VincentyGeodesy.MoveInDirection(VelocityDBExtensions.Geo.WGS84Point,System.Double,System.Double)"]

bearingInDegrees

Type: [System.Double](#)

[Missing <param name="bearingInDegrees"/> documentation for "M:VelocityDBExtensions.Geo.util.VincentyGeodesy.MoveInDirection(VelocityDBExtensions.Geo.WGS84Point,System.Double,System.Double)"]

distanceInMeters

Type: [System.Double](#)

[Missing <param name="distanceInMeters"/> documentation for "M:VelocityDBExtensions.Geo.util.VincentyGeodesy.MoveInDirection(VelocityDBExtensions.Geo.WGS84Point,System.Double,System.Double)"]

Return Value

Type: [WGS84Point](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Geo.util.VincentyGeodesy.MoveInDirection(VelocityDBExtensions.Geo.WGS84Point,System.Double,System.Double)"]

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.VincentyGeodesy Fields

The [VincentyGeodesy](#) type exposes the following members.

Fields

| | Name | Description |
|--------------------------------------------------------------------------------------------|--------------------------|-------------|
|  S | DegToRad | |
|  S | EPSILON | |

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.DegToRad Field

[Missing <summary> documentation for "F:VelocityDBExtensions.Geo.util.VincentyGeodesy.DegToRad"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const double DegToRad = 0.0174532925199433
```

VB

```
Public Const DegToRad As Double = 0.0174532925199433
```

C++

```
public:  
literal double DegToRad = 0.0174532925199433
```

F#

```
static val mutable DegToRad: float
```

Field Value

Type: [Double](#)

See Also

[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)

VincentyGeodesy.EPSILON Field

[Missing <summary> documentation for "F:VelocityDBExtensions.Geo.util.VincentyGeodesy.EPSILON"]

Namespace: [VelocityDBExtensions.Geo.util](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const double EPSILON = 1E-12
```

VB

```
Public Const EPSILON As Double = 1E-12
```

C++

```
public:  
literal double EPSILON = 1E-12
```

F#

```
static val mutable EPSILON: float
```

Field Value

Type: [Double](#)

See Also




[VincentyGeodesy Class](#)

[VelocityDBExtensions.Geo.util Namespace](#)



VelocityDBExtensions.Spatial Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions.Spatial"]

Classes

| Class | Description |
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  PriorityQueue(Priority, Value) | Priority Queue that stores values and priorities. |
|  PriorityQueueRTree | RTree uses double for priorities and Rectangle as values |
|  RTree | RTree implementation inspired by description on Wikipedia http://en.wikipedia.org/wiki/R-tree |

Structures

| Structure | Description |
|--------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Point | Points are most often considered within the framework of Euclidean geometry, where they are one of the fundamental objects. Euclid originally defined the point as "that which has no part". In two-dimensional Euclidean space, a point is represented by an ordered pair (x,y) of numbers, where the first number conventionally represents the horizontal and is often denoted by x, and the second number conventionally represents the vertical and is often denoted by y. |
|  Rectangle | |

Point Structure

Points are most often considered within the framework of Euclidean geometry, where they are one of the fundamental objects. Euclid originally defined the point as "that which has no part". In two-dimensional Euclidean space, a point is represented by an ordered pair (x,y) of numbers, where the first number conventionally represents the horizontal and is often denoted by x, and the second number conventionally represents the vertical and is often denoted by y.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public struct Point
```

VB

```
Public Structure Point
```

C++


```
public value class Point
```

F#

```
[<SealedAttribute>]
type Point = struct end
```

The **Point** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|--------------------------|
|  | Point | Creates a Point instance |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------------|
|  | ToString | return "(" + x + ", " + y + ")"; (Overrides ValueType.ToString() .) |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------|--------------------------------------|
|  | x | The (x, y) coordinates of the point. |
|  | y | The (x, y) coordinates of the point. |

See Also

[VelocityDBExtensions.Spatial Namespace](#)

Point Constructor

Creates a Point instance

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Point(  
    double x,  
    double y  
)
```

VB

```
Public Sub New (  
    x As Double,  
    y As Double  
)
```

C++

```
public:  
Point(  
    double x,  
    double y  
)
```

F#

```
new :  
    x : float *  
    y : float -> Point
```

Parameters

x

Type: [System.Double](#)

The x coordinate of the point

y

Type: [System.Double](#)

The y coordinate of the point

See Also


[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Point.Point Methods

The [Point](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------------|
|  | ToString | return "(" + x + ", " + y + ")"; (Overrides ValueType.ToString() .) |

See Also

[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Point.ToString Method

```
return "(" + x + ", " + y + ")";
```

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Point.ToString"]

See Also



[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Point.Point Fields

The [Point](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------|--------------------------------------|
|  | x | The (x, y) coordinates of the point. |
|  | y | The (x, y) coordinates of the point. |

See Also

[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Point.x Field

The (x, y) coordinates of the point.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double x
```

VB

```
Public x As Double
```

C++

```
public:  
double x
```

F#

```
val mutable x: float
```

Field Value

Type: [Double](#)

See Also

[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Point.y Field

The (x, y) coordinates of the point.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double y
```

VB

```
Public y As Double
```

C++

```
public:  
double y
```

F#

```
val mutable y: float
```

Field Value

Type: [Double](#)

See Also

[Point Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value) Class

Priority Queue that stores values and priorities.

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions.Spatial.PriorityQueue(Priority, Value)

[VelocityDBExtensions.Spatial.PriorityQueueRTree](#)

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class PriorityQueue<Priority, Value>  
where Priority : IComparable
```

VB

```
Public Class PriorityQueue (Of Priority As IComparable, Value)
```

C++

```
generic<typename Priority, typename Value>  
where Priority : IComparable  
public ref class PriorityQueue
```

F#

```
type PriorityQueue<'Priority, 'Value when 'Priority : IComparable> = class  
end
```

Type Parameters

Priority

[Missing <typeparam name="Priority"/> documentation for "T:VelocityDBExtensions.Spatial.PriorityQueue`2"]

Value





[Missing <typeparam name="Value"/> documentation for "T:VelocityDBExtensions.Spatial.PriorityQueue`2"]

The PriorityQueue(Priority, Value) type exposes the following members.




Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------|-----------------------|
|  | PriorityQueue(Priority, Value) | Creates a PriorityQue |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|---------------------------------------------------------------------|
|  | Count | The number of values and priorities that are in the que |
|  | PriorityPeek | Peek at the next Priority |
|  | SortOrderAscending | Set or get sorting order, ascending if true or descending if false. |
|  | ValuePeek | Peek at the next value |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Clear | Removes all values and priorities from the que |
|  | Insert | Insert a value, append it to the arrays, then reheapify by promoting it to the correct place. |
|  | Pop | Get the value with the lowest priority creates a "hole" at the root of the tree. The algorithm swaps the hole with the appropriate child, until the last entry will fit correctly into the hole (ie is lower priority than its children) |

See Also

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(*Priority, Value*) Constructor

Creates a PriorityQue

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public PriorityQueue(  
    bool sortOrderAscending = true,  
    int initialCapacity = 25  
)
```

VB

```
Public Sub New (  
    Optional sortOrderAscending As Boolean = true,  
    Optional initialCapacity As Integer = 25  
)
```

C++

```
public:  
PriorityQueue(  
    bool sortOrderAscending = true,  
    int initialCapacity = 25  
)
```

F#

```
new :  
    ?sortOrderAscending : bool *  
    ?initialCapacity : int  
(* Defaults:  
    let _sortOrderAscending = defaultArg sortOrderAscending true  
    let _initialCapacity = defaultArg initialCapacity 25  
)  
-> PriorityQueue
```

Parameters

sortOrderAscending (Optional)

Type: [System.Boolean](#)

order ascending if true otherwise order descending

initialCapacity (Optional)

Type: [System.Int32](#)

improve performance by setting expected size initially





See Also

[PriorityQueue\(Priority, Value\)Class](#)

PriorityQueue(Priority, Value).PriorityQueue(Priority, Value) Properties

The [PriorityQueue\(Priority, Value\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|---------------------------------------------------------------------|
|  | Count | The number of values and priorities that are in the que |
|  | PriorityPeek | Peek at the next Priority |
|  | SortOrderAscending | Set or get sorting order, ascending if true or descending if false. |
|  | ValuePeek | Peek at the next value |

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).Count Property

The number of values and priorities that are in the que

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual int Count { get; }
```

VB

```
Public Overridable ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get ();  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).PriorityPeek Property

Peek at the next Priority

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Priority PriorityPeek { get; }
```

VB

```
Public ReadOnly Property PriorityPeek As Priority  
    Get
```

C++

```
public:  
property Priority PriorityPeek {  
    Priority get ();  
}
```

F#

```
member PriorityPeek : 'Priority with get
```

Property Value

Type: *Priority*

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).SortOrderAscending Property

Set or get sorting order, ascending if true or descending if false.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual bool SortOrderAscending { get; set; }
```

VB

```
Public Overridable Property SortOrderAscending As Boolean  
    Get  
    Set
```

C++

```
public:  
virtual property bool SortOrderAscending {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
abstract SortOrderAscending : bool with get, set  
override SortOrderAscending : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).ValuePeek Property

Peek at the next value

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Value ValuePeek { get; }
```

VB

```
Public ReadOnly Property ValuePeek As Value  
    Get
```

C++

```
public:  
property Value ValuePeek {  
    Value get ();  
}
```

F#

```
member ValuePeek : 'Value with get
```

Property Value

Type: *Value*

See Also




[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).PriorityQueue(Priority, Value) Methods

The [PriorityQueue\(Priority, Value\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Clear | Removes all values and priorities from the que |
|  | Insert | Insert a value, append it to the arrays, then reheapify by promoting it to the correct place. |
|  | Pop | Get the value with the lowest priority creates a "hole" at the root of the tree. The algorithm swaps the hole with the appropriate child, until the last entry will fit correctly into the hole (ie is lower priority than its children) |

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).Clear Method

Removes all values and priorities from the que

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Clear ()
```

VB

```
Public Sub Clear
```

C++

```
public:  
void Clear ()
```

F#

```
member Clear : unit -> unit
```

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).Insert Method

Insert a value, append it to the arrays, then reheapify by promoting it to the correct place.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual void Insert(  
    Value value,  
    Priority priority  
)
```

VB

```
Public Overridable Sub Insert (  
    value As Value,  
    priority As Priority  
)
```

C++

```
public:  
virtual void Insert(  
    Value value,  
    Priority priority  
)
```

F#

```
abstract Insert :  
    value : 'Value *  
    priority : 'Priority -> unit  
override Insert :  
    value : 'Value *  
    priority : 'Priority -> unit
```

Parameters

value

Type: *Value*

the value inserted

priority

Type: *Priority*

the priority of the value inserted

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueue(Priority, Value).Pop Method

Get the value with the lowest priority creates a "hole" at the root of the tree. The algorithm swaps the hole with the appropriate child, until the last entry will fit correctly into the hole (ie is lower priority than its children)

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public virtual Value Pop()
```

VB

```
Public Overridable Function Pop As Value
```

C++

```
public:  
virtual Value Pop()
```

F#

```
abstract Pop : unit -> 'Value  
override Pop : unit -> 'Value
```

Return Value

Type: *Value*

the Value with the lowest priority

See Also

[PriorityQueue\(Priority, Value\)Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueueRTree Class

RTree uses double for priorities and Rectangle as values

Inheritance Hierarchy

[System.Object](#)

[VelocityDBExtensions.Spatial.PriorityQueue\(Double, Rectangle\)](#)

VelocityDBExtensions.Spatial.PriorityQueueRTree

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class PriorityQueueRTree : PriorityQueue<double, Rectangle>
```

VB

```
Public Class PriorityQueueRTree  
    Inherits PriorityQueue(Of Double, Rectangle)
```

C++


```
public ref class PriorityQueueRTree : public PriorityQueue<double, Rectangle>
```

F#

```
type PriorityQueueRTree =  
    class  
        inherit PriorityQueue<float, Rectangle>  
    end
```

The **PriorityQueueRTree** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------|
|  | PriorityQueueRTree | Initializes a new instance of the PriorityQueueRTree class |

See Also

[VelocityDBExtensions.Spatial Namespace](#)

PriorityQueueRTree Constructor

Initializes a new instance of the [PriorityQueueRTree](#) class

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public PriorityQueueRTree(  
    bool sortOrderAscending = true,  
    int initialCapacity = 25  
)
```

VB

```
Public Sub New (  
    Optional sortOrderAscending As Boolean = true,  
    Optional initialCapacity As Integer = 25  
)
```

C++

```
public:  
PriorityQueueRTree (  
    bool sortOrderAscending = true,  
    int initialCapacity = 25  
)
```

F#

```
new :  
    ?sortOrderAscending : bool *  
    ?initialCapacity : int  
(* Defaults:  
    let _sortOrderAscending = defaultArg sortOrderAscending true  
    let _initialCapacity = defaultArg initialCapacity 25  
)  
-> PriorityQueueRTree
```

Parameters

sortOrderAscending (Optional)

Type: [System.Boolean](#)

[Missing <param name="sortOrderAscending"/> documentation for "M:VelocityDBExtensions.Spatial.PriorityQueueRTree.#ctor(System.Boolean,System.Int32)"]

initialCapacity (Optional)

Type: [System.Int32](#)

[Missing <param name="initialCapacity"/> documentation for "M:VelocityDBExtensions.Spatial.PriorityQueueRTree.#ctor(System.Boolean,System.Int32)"]

VelocityDB Class Library

See Also

[PriorityQueueRTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle Structure

[Missing <summary> documentation for "T:VelocityDBExtensions.Spatial.Rectangle"]

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public struct Rectangle : IComparable
```

VB

```
Public Structure Rectangle
    Implements IComparable
```

C++


```
public value class Rectangle : IComparable
```

F#










```
[<SealedAttribute>]
type Rectangle =
    struct
        interface IComparable
    end
```

The **Rectangle** type exposes the following members.

















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------|
|  | Rectangle | Initializes a new instance of the Rectangle class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------|
|  | Area | Compute the area of this rectangle. |
|  | AspectRatio | The aspect ratio denotes the ratio of length to width of the rectangle (Width / Height) |
|  | Centre | Gets the centre of the rectangle |
|  | Height | The height of a rectangle |
|  | MaxX | The high x coordinate |
|  | MaxY | The high y coordinate |
|  | MinX | The low x coordinate |
|  | MinY | The low y coordinate |
|  | Width | The width of a rectangle |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | add | Computes the union of this rectangle and the passed point, storing the result in this rectangle. |
|  | Add | Computes the union of this rectangle and the passed rectangle, storing the result in this rectangle. |
|  | CompareTo | |
|  | containedBy | Determine whether this rectangle is contained by the passed rectangle |
|  | Contains | Determine whether this rectangle contains the passed rectangle |
|  | distance | Return the distance between a rectangle and a point. If the rectangle contains the point, the distance is zero. |
|  | Distance(Point) | Return the distance between this rectangle and the passed point. If the rectangle contains the point, the distance is zero. |
|  | Distance(Rectangle) | Return the distance between this rectangle and the passed rectangle. If the rectangles overlap, the distance is zero. |
|  | distanceSq | Get the square of the distance between two points. |
|  | edgeOverlaps | Determine whether an edge of this rectangle overlies the equivalent edge of the passed rectangle |
|   | enlargement | Calculate the area by which a rectangle would be enlarged if added to the passed rectangle |
|  | Enlargement | Calculate the area by which this rectangle would be enlarged if added to the passed rectangle. Neither rectangle is altered. |
|  | Equals | |
|  | GetHashCode | Customized hash code using the coordinates of the rectangle $37 * \text{minX} * \text{minY} * \text{maxX} * \text{maxY}$ (Overrides ValueType.GetHashCode() .) |
|  | Intersects | Determine whether this rectangle intersects the passed rectangle |
|  | sameObject | Determine whether this rectangle is the same as another object. Note that two rectangles can be equal but not the same object, if they both have the same bounds. |
|  | set | Sets the size of this rectangle to equal the passed rectangle. |
|  | ToString | Return a string representation of this rectangle, in the form: (1.2, 3.4), (5.6, 7.8) (Overrides ValueType.ToString() .) |
|  | union | Find the the union of this rectangle and the passed rectangle. Neither rectangle is altered |

See Also

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle Constructor

Initializes a new instance of the [Rectangle](#) class

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Rectangle(  
    double x1,  
    double y1,  
    double x2,  
    double y2  
)
```

VB

```
Public Sub New (  
    x1 As Double,  
    y1 As Double,  
    x2 As Double,  
    y2 As Double  
)
```

C++

```
public:  
Rectangle(  
    double x1,  
    double y1,  
    double x2,  
    double y2  
)
```

F#

```
new :  
    x1 : float *  
    y1 : float *  
    x2 : float *  
    y2 : float -> Rectangle
```

Parameters

x1

Type: [System.Double](#)

coordinate of any corner of the rectangle

y1

Type: [System.Double](#)

(see *x1*)

VelocityDB Class Library

x2

Type: [System.Double](#)

coordinate of the opposite corner

y2

Type: [System.Double](#)

(see x2)

See Also










[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Rectangle Properties

The [Rectangle](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------------------------------------------|
|  | Area | Compute the area of this rectangle. |
|  | AspectRatio | The aspect ratio denotes the ratio of length to width of the rectangle (Width / Height) |
|  | Centre | Gets the centre of the rectangle |
|  | Height | The height of a rectangle |
|  | MaxX | The high x coordinate |
|  | MaxY | The high y coordinate |
|  | MinX | The low x coordinate |
|  | MinY | The low y coordinate |
|  | Width | The width of a rectangle |

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Area Property

Compute the area of this rectangle.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Area { get; }
```

VB

```
Public ReadOnly Property Area As Double  
    Get
```

C++

```
public:  
property double Area {  
    double get ();  
}
```

F#

```
member Area : float with get
```

Return Value

Type: [Double](#)

The area of this rectangle

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.AspectRatio Property

The aspect ratio denotes the ratio of length to width of the rectangle (Width / Height)

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double AspectRatio { get; }
```

VB

```
Public ReadOnly Property AspectRatio As Double  
    Get
```

C++

```
public:  
property double AspectRatio {  
    double get ();  
}
```

F#

```
member AspectRatio : float with get
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Centre Property

Gets the centre of the rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Point Centre { get; }
```

VB

```
Public ReadOnly Property Centre As Point  
    Get
```

C++

```
public:  
property Point Centre {  
    Point get ();  
}
```

F#

```
member Centre : Point with get
```

Property Value

Type: [Point](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Height Property

The height of a rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Height { get; }
```

VB

```
Public ReadOnly Property Height As Double  
    Get
```

C++

```
public:  
property double Height {  
    double get ();  
}
```

F#

```
member Height : float with get
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.MaxX Property

The high x coordinate

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double MaxX { get; set; }
```

VB

```
Public Property MaxX As Double  
    Get  
    Set
```

C++

```
public:  
property double MaxX {  
    double get ();  
    void set (double value);  
}
```

F#

```
member MaxX : float with get, set
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.MaxY Property

The high y coordinate

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double MaxY { get; set; }
```

VB

```
Public Property MaxY As Double  
    Get  
    Set
```

C++

```
public:  
property double MaxY {  
    double get ();  
    void set (double value);  
}
```

F#

```
member MaxY : float with get, set
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.MinX Property

The low x coordinate

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double MinX { get; set; }
```

VB

```
Public Property MinX As Double  
    Get  
    Set
```

C++

```
public:  
property double MinX {  
    double get ();  
    void set (double value);  
}
```

F#

```
member MinX : float with get, set
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.MinY Property

The low y coordinate

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double MinY { get; set; }
```

VB

```
Public Property MinY As Double  
    Get  
    Set
```

C++

```
public:  
property double MinY {  
    double get ();  
    void set (double value);  
}
```

F#

```
member MinY : float with get, set
```

Property Value

Type: [Double](#)

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Width Property

The width of a rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Width { get; }
```

VB

```
Public ReadOnly Property Width As Double  
    Get
```

C++

```
public:  
property double Width {  
    double get ();  
}
```

F#

```
member Width : float with get
```

Property Value

Type: [Double](#)

See Also





[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Rectangle Methods

The [Rectangle](#) type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  add | Computes the union of this rectangle and the passed point, storing the result in this rectangle. |
|  Add | Computes the union of this rectangle and the passed rectangle, storing the result in this rectangle. |
|  CompareTo | |
|  containedBy | Determine whether this rectangle is contained by the passed rectangle |
|  Contains | Determine whether this rectangle contains the passed rectangle |
|  distance | Return the distance between a rectangle and a point. If the rectangle contains the point, the distance is zero. |
|  Distance(Point) | Return the distance between this rectangle and the passed point. If the rectangle contains the point, the distance is zero. |
|  Distance(Rectangle) | Return the distance between this rectangle and the passed rectangle. If the rectangles overlap, the distance is zero. |
|  distanceSq | Get the square of the distance between two points. |
|  edgeOverlaps | Determine whether an edge of this rectangle overlies the equivalent edge of the passed rectangle |
|  enlargement | Calculate the area by which a rectangle would be enlarged if added to the passed rectangle |
|  Enlargement | Calculate the area by which this rectangle would be enlarged if added to the passed rectangle. Neither rectangle is altered. |
|  Equals | |
|  GetHashCode | Customized hash code using the coordinates of the rectangle $37 * \text{minX} * \text{minY} * \text{maxX} * \text{maxY}$ (Overrides ValueType.GetHashCode() .) |
|  Intersects | Determine whether this rectangle intersects the passed rectangle |
|  sameObject | Determine whether this rectangle is the same as another object. Note that two rectangles can be equal but not the same object, if they both have the same bounds. |
|  set | Sets the size of this rectangle to equal the passed rectangle. |
|  ToString | Return a string representation of this rectangle, in the form: (1.2, 3.4), (5.6, 7.8) (Overrides ValueType.ToString() .) |
|  union | Find the the union of this rectangle and the passed rectangle. Neither rectangle is altered |

VelocityDB Class Library

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.add Method

Computes the union of this rectangle and the passed point, storing the result in this rectangle.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void add(  
    Point p  
)
```

VB

```
Public Sub add (  
    p As Point  
)
```

C++

```
public:  
void add(  
    Point p  
)
```

F#

```
member add :  
    p : Point -> unit
```

Parameters

p

Type: [VelocityDBExtensions.Spatial.Point](#)

Point to add to this rectangle

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Add Method

Computes the union of this rectangle and the passed rectangle, storing the result in this rectangle.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Add(  
    Rectangle r  
)
```

VB

```
Public Sub Add (  
    r As Rectangle  
)
```

C++

```
public:  
void Add(  
    Rectangle r  
)
```

F#

```
member Add :  
    r : Rectangle -> unit
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

Rectangle to add to this rectangle

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.CompareTo Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.CompareTo(System.Object)"]

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int CompareTo(  
    Object o  
)
```

VB

```
Public Function CompareTo (  
    o As Object  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    Object^ o  
) sealed
```

F#

```
abstract CompareTo :  
    o : Object -> int  
override CompareTo :  
    o : Object -> int
```

Parameters

o

Type: [System.Object](#)

[Missing <param name="o"/> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.CompareTo(System.Object)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.CompareTo(System.Object)"]

Implements

[IComparable.CompareTo\(Object\)](#)

VelocityDB Class Library

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.containedBy Method

Determine whether this rectangle is contained by the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool containedBy(  
    Rectangle r  
)
```

VB

```
Public Function containedBy (  
    r As Rectangle  
) As Boolean
```

C++

```
public:  
bool containedBy(  
    Rectangle r  
)
```

F#

```
member containedBy :  
    r : Rectangle -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

The rectangle that might contain this rectangle

Return Value

Type: [Boolean](#)

return true if the passed rectangle contains this rectangle, false if it does not

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Contains Method

Determine whether this rectangle contains the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Contains(  
    Rectangle r  
)
```

VB

```
Public Function Contains (  
    r As Rectangle  
) As Boolean
```

C++

```
public:  
bool Contains(  
    Rectangle r  
)
```

F#

```
member Contains :  
    r : Rectangle -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

The rectangle that might be contained by this rectangle

Return Value

Type: [Boolean](#)

return true if this rectangle contains the passed rectangle, false if it does not

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.distance Method

Return the distance between a rectangle and a point. If the rectangle contains the point, the distance is zero.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double distance(  
    double pX,  
    double pY  
)
```

VB

```
Public Function distance (  
    pX As Double,  
    pY As Double  
) As Double
```

C++

```
public:  
double distance(  
    double pX,  
    double pY  
)
```

F#

```
member distance :  
    pX : float *  
    pY : float -> float
```

Parameters

pX

Type: [System.Double](#)

X coordinate of point

pY

Type: [System.Double](#)

Y coordinate of point

Return Value

Type: [Double](#)

return distance between this rectangle and the passed point.

VelocityDB Class Library



See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Distance Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
|  | Distance(Point) | Return the distance between this rectangle and the passed point. If the rectangle contains the point, the distance is zero. |
|  | Distance(Rectangle) | Return the distance between this rectangle and the passed rectangle. If the rectangles overlap, the distance is zero. |

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Distance Method (Point)

Return the distance between this rectangle and the passed point. If the rectangle contains the point, the distance is zero.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Distance(  
    Point p  
)
```

VB

```
Public Function Distance (  
    p As Point  
) As Double
```

C++

```
public:  
double Distance(  
    Point p  
)
```

F#

```
member Distance :  
    p : Point -> float
```

Parameters

p

Type: [VelocityDBExtensions.Spatial.Point](#)

Point to find the distance to

Return Value

Type: [Double](#)

return distance between this rectangle and the passed point.

See Also

[Rectangle Structure](#)

[Distance Overload](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Distance Method (Rectangle)

Return the distance between this rectangle and the passed rectangle. If the rectangles overlap, the distance is zero.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Distance(  
    Rectangle r  
)
```

VB

```
Public Function Distance (  
    r As Rectangle  
) As Double
```

C++

```
public:  
double Distance(  
    Rectangle r  
)
```

F#

```
member Distance :  
    r : Rectangle -> float
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

Rectangle to find the distance to

Return Value

Type: [Double](#)

return distance between this rectangle and the passed rectangle

See Also

[Rectangle Structure](#)

[Distance Overload](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.distanceSq Method

Get the square of the distance between two points.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double distanceSq(  
    double pX,  
    double pY  
)
```

VB

```
Public Function distanceSq (  
    pX As Double,  
    pY As Double  
) As Double
```

C++

```
public:  
double distanceSq(  
    double pX,  
    double pY  
)
```

F#

```
member distanceSq :  
    pX : float *  
    pY : float -> float
```

Parameters

pX

Type: [System.Double](#)

the x coordinate of point

pY

Type: [System.Double](#)

the y coordinate of point

Return Value

Type: [Double](#)

the square of the distance between two points.

See Also

[Rectangle Structure](#)

Rectangle.edgeOverlaps Method

Determine whether an edge of this rectangle overlies the equivalent edge of the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool edgeOverlaps(  
    Rectangle r  
)
```

VB

```
Public Function edgeOverlaps (  
    r As Rectangle  
) As Boolean
```

C++

```
public:  
bool edgeOverlaps(  
    Rectangle r  
)
```

F#

```
member edgeOverlaps :  
    r : Rectangle -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

[Missing <param name="r"/> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.edgeOverlaps(VelocityDBExtensions.Spatial.Rectangle)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.edgeOverlaps(VelocityDBExtensions.Spatial.Rectangle)"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.enlargement Method

Calculate the area by which a rectangle would be enlarged if added to the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

```
C#  
public static double enlargement(  
    Rectangle r1,  
    Rectangle r2  
)
```

```
VB  
Public Shared Function enlargement (  
    r1 As Rectangle,  
    r2 As Rectangle  
) As Double
```

```
C++  
public:  
static double enlargement(  
    Rectangle r1,  
    Rectangle r2  
)
```

```
F#  
static member enlargement :  
    r1 : Rectangle *  
    r2 : Rectangle -> float
```

Parameters

r1

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

minimum X coordinate of rectangle 1

r2

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

minimum X coordinate of rectangle 2

Return Value

Type: [Double](#)

return enlargement

See Also

[Rectangle Structure](#)

Rectangle.Enlargement Method

Calculate the area by which this rectangle would be enlarged if added to the passed rectangle. Neither rectangle is altered.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public double Enlargement(  
    ref Rectangle r  
)
```

VB

```
Public Function Enlargement (  
    ByRef r As Rectangle  
) As Double
```

C++

```
public:  
double Enlargement(  
    Rectangle% r  
)
```

F#

```
member Enlargement :  
    r : Rectangle byref -> float
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

Rectangle to union with this rectangle, in order to compute the difference in area of the union and the original rectangle

Return Value

Type: [Double](#)

enlargement

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Equals Method

[Missing <summary> documentation for

"M:VelocityDBExtensions.Spatial.Rectangle.Equals(VelocityDBExtensions.Spatial.Rectangle)"]

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Equals(  
    Rectangle r  
)
```

VB

```
Public Function Equals (  
    r As Rectangle  
) As Boolean
```

C++

```
public:  
bool Equals(  
    Rectangle r  
)
```

F#

```
member Equals :  
    r : Rectangle -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

[Missing <param name="r"/> documentation for

"M:VelocityDBExtensions.Spatial.Rectangle.Equals(VelocityDBExtensions.Spatial.Rectangle)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions.Spatial.Rectangle.Equals(VelocityDBExtensions.Spatial.Rectangle)"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.GetHashCode Method

Customized hash code using the coordinates of the rectangle $37 * \text{minX} * \text{minY} * \text{maxX} * \text{maxY}$

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.GetHashCode"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.Intersects Method

Determine whether this rectangle intersects the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Intersects(  
    ref Rectangle? r  
)
```

VB

```
Public Function Intersects (  
    ByRef r As Rectangle?  
) As Boolean
```

C++

```
public:  
bool Intersects(  
    Nullable<Rectangle>% r  
)
```

F#

```
member Intersects :  
    r : Nullable<Rectangle> byref -> bool
```

Parameters

r

Type: [System.Nullable\(Rectangle\)](#)

The rectangle that might intersect this rectangle

Return Value

Type: [Boolean](#)

return true if the rectangles intersect, false if they do not intersect

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.sameObject Method

Determine whether this rectangle is the same as another object. Note that two rectangles can be equal but not the same object, if they both have the same bounds.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool sameObject(  
    Object o  
)
```

VB

```
Public Function sameObject (  
    o As Object  
) As Boolean
```

C++

```
public:  
bool sameObject(  
    Object^ o  
)
```

F#

```
member sameObject :  
    o : Object -> bool
```

Parameters

o

Type: [System.Object](#)

The object to compare with this rectangle

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.sameObject(System.Object)"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.set Method

Sets the size of this rectangle to equal the passed rectangle.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void set (  
    Rectangle r  
)
```

VB

```
Public Sub set (  
    r As Rectangle  
)
```

C++

```
public:  
void set (  
    Rectangle r  
)
```

F#

```
member set :  
    r : Rectangle -> unit
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

[Missing <param name="r"/> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.set(VelocityDBExtensions.Spatial.Rectangle)"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.ToString Method

Return a string representation of this rectangle, in the form: (1.2, 3.4), (5.6, 7.8)

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

return String String representation of this rectangle

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

Rectangle.union Method

Find the the union of this rectangle and the passed rectangle. Neither rectangle is altered

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Rectangle union(  
    Rectangle r  
)
```

VB

```
Public Function union (  
    r As Rectangle  
) As Rectangle
```

C++

```
public:  
Rectangle union(  
    Rectangle r  
)
```

F#

```
member union :  
    r : Rectangle -> Rectangle
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

The rectangle to union with this rectangle

Return Value

Type: [Rectangle](#)

[Missing <returns> documentation for "M:VelocityDBExtensions.Spatial.Rectangle.union(VelocityDBExtensions.Spatial.Rectangle)"]

See Also

[Rectangle Structure](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree Class

RTree implementation inspired by description on Wikipedia <http://en.wikipedia.org/wiki/R-tree>

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions.Spatial.RTree

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class RTree : OptimizedPersistable
```

VB

```
Public Class RTree
    Inherits OptimizedPersistable
```

C++


```
public ref class RTree : public OptimizedPersistable
```

F#



```
type RTree =
    class
        inherit OptimizedPersistable
    end
```

The **RTree** type exposes the following members.









Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | RTree | RTree implementaion customized for optimal performance with VelocityDB. Follows outline as described in http://en.wikipedia.org/wiki/R-tree |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------|
|  | Bounds | Gets the bounds of all the entries in the spatial index, or null if there are no entries. |
|  | Count | Get the number of rectangles that are managed by the Rtree |



Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Add | Start at the Root Node Select the child that needs the least enlargement in order to fit the new geometry. Repeat until at a leaf node. If leaf node has available space insert Else split the entry into two nodes Update parent nodes Update the entry that pointed to the node with a new minimum bounding rectangle Add a new entry for the second new node If there is no space in the parent node, split and repeat |
|  | checkConsistency | Check the consistency of the tree. |
|  | Contains | Finds all rectangles contained by the passed rectangle |
|  | InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  | Intersects | Finds all rectangles that intersect the passed rectangle. |
|  | Nearest | Finds the nearest rectangles to the passed point. If multiple rectangles are equally near, they will all be returned. |
|  | NearestN | Finds the N nearest rectangles to the passed point. If multiple rectangles are equally near, they will all (but total limited to N) be returned. |
|  | Remove | Removes a rectangle from the Rtree |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|------------------------------------------------------------|
|  | defaultMaxNodeEntries | default maximum number of reactangles in a leaf RTree node |
|  | defaultMinNodeEntries | default minimum number of reactangles in a leaf RTree node |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDBExtensions.Spatial Namespace](#)

RTree Constructor

RTree implementaion customized for optimal performance with VelocityDB. Follows outline as described in <http://en.wikipedia.org/wiki/R-tree>

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public RTree(  
    int minNodeEntries = 80,  
    int maxNodeEntries = 200  
)
```

VB

```
Public Sub New (  
    Optional minNodeEntries As Integer = 80,  
    Optional maxNodeEntries As Integer = 200  
)
```

C++

```
public:  
RTree(  
    int minNodeEntries = 80,  
    int maxNodeEntries = 200  
)
```

F#

```
new :  
    ?minNodeEntries : int *  
    ?maxNodeEntries : int  
(* Defaults:  
    let _minNodeEntries = defaultArg minNodeEntries 80  
    let _maxNodeEntries = defaultArg maxNodeEntries 200  
*)  
-> RTree
```

Parameters

minNodeEntries (Optional)

Type: [System.Int32](#)

Minimum number of entries in a node. The default value is 80.

maxNodeEntries (Optional)

Type: [System.Int32](#)

Maximum number of entries in a node. The default value is 200.

VelocityDB Class Library

See Also



[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.RTree Properties

The [RTree](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------------------------------------------------------------------------------------|
|  | Bounds | Gets the bounds of all the entries in the spatial index, or null if there are no entries. |
|  | Count | Get the number of rectangles that are managed by the Rtree |

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Bounds Property

Gets the bounds of all the entries in the spatial index, or null if there are no entries.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Rectangle? Bounds { get; }
```

VB

```
Public ReadOnly Property Bounds As Rectangle?  
    Get
```

C++

```
public:  
property Nullable<Rectangle> Bounds {  
    Nullable<Rectangle> get ();  
}
```

F#

```
member Bounds : Nullable<Rectangle> with get
```

Property Value

Type: [Nullable\(Rectangle\)](#)

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Count Property

Get the number of rectangles that are managed by the Rtree

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int Count { get; }
```

VB

```
Public ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
property int Count {  
    int get ();  
}
```

F#

```
member Count : int with get
```

Property Value

Type: [Int32](#)

See Also









[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)



RTree.RTree Methods

The [RTree](#) type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Add | Start at the Root Node Select the child that needs the least enlargement in order to fit the new geometry. Repeat until at a leaf node. If leaf node has available space insert Else split the entry into two nodes Update parent nodes Update the entry that pointed to the node with a new minimum bounding rectangle Add a new entry for the second new node If there is no space in the parent node, split and repeat |
|  checkConsistency | Check the consistency of the tree. |
|  Contains | Finds all rectangles contained by the passed rectangle |
|  InitializeAfterRead | (Overrides OptimizedPersistable.InitializeAfterRead(SessionBase).) |
|  Intersects | Finds all rectangles that intersect the passed rectangle. |
|  Nearest | Finds the nearest rectangles to the passed point. If multiple rectangles are equally near, they will all be returned. |
|  NearestN | Finds the N nearest rectangles to the passed point. If multiple rectangles are equally near, they will all (but total limited to N) be returned. |
|  Remove | Removes a rectangle from the Rtree |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Add Method

Start at the Root Node Select the child that needs the least enlargement in order to fit the new geometry. Repeat until at a leaf node. If leaf node has available space insert Else split the entry into two nodes Update parent nodes Update the entry that pointed to the node with a new minimum bounding rectangle Add a new entry for the second new node If there is no space in the parent node, split and repeat

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Add(  
    Rectangle r  
)
```

VB

```
Public Sub Add (  
    r As Rectangle  
)
```

C++

```
public:  
void Add(  
    Rectangle r  
)
```

F#

```
member Add :  
    r : Rectangle -> unit
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)
the rectangle being added

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.checkConsistency Method

Check the consistency of the tree.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool checkConsistency()
```

VB

```
Public Function checkConsistency As Boolean
```

C++

```
public:  
bool checkConsistency()
```

F#

```
member checkConsistency : unit -> bool
```

Return Value

Type: [Boolean](#)

false if an inconsistency is detected, true otherwise

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Contains Method

Finds all rectangles contained by the passed rectangle

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Contains(  
    Rectangle r,  
    Func<Rectangle, bool> v  
)
```

VB

```
Public Sub Contains (  
    r As Rectangle,  
    v As Func(Of Rectangle, Boolean)  
)
```

C++

```
public:  
void Contains(  
    Rectangle r,  
    Func<Rectangle, bool>^ v  
)
```

F#

```
member Contains :  
    r : Rectangle *  
    v : Func<Rectangle, bool> -> unit
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

The rectangle for which this method finds contained rectangles.

v

Type: [System.Func\(Rectangle, Boolean\)](#)

if return true, continue seach

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.InitializeAfterRead Method

[Missing <summary> documentation for "M:VelocityDBExtensions.Spatial.RTree.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void InitializeAfterRead(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub InitializeAfterRead (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void InitializeAfterRead(  
    SessionBase^ session  
) override
```

F#

```
abstract InitializeAfterRead :  
    session : SessionBase -> unit  
override InitializeAfterRead :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions.Spatial.RTree.InitializeAfterRead(VelocityDb.Session.SessionBase)"]

Implements

[IOptimizedPersistable.InitializeAfterRead\(SessionBase\)](#)

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Intersects Method

Finds all rectangles that intersect the passed rectangle.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Intersects(  
    Rectangle r,  
    Func<Rectangle, bool> v  
)
```

VB

```
Public Function Intersects (  
    r As Rectangle,  
    v As Func(Of Rectangle, Boolean)  
) As Boolean
```

C++

```
public:  
bool Intersects(  
    Rectangle r,  
    Func<Rectangle, bool>^ v  
)
```

F#

```
member Intersects :  
    r : Rectangle *  
    v : Func<Rectangle, bool> -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

the rectangle we are intersecting with

v

Type: [System.Func\(Rectangle, Boolean\)](#)

if returns true, search continues else search is ended

Return Value

Type: [Boolean](#)

true if at least one intersection was found and v returns true

See Also

[RTree Class](#)

RTree.Nearest Method

Finds the nearest rectangles to the passed point. If multiple rectangles are equally near, they will all be returned.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public PriorityQueueRTree Nearest(  
    Point p,  
    double furthestDistance  
)
```

VB

```
Public Function Nearest (  
    p As Point,  
    furthestDistance As Double  
) As PriorityQueueRTree
```

C++

```
public:  
PriorityQueueRTree^ Nearest(  
    Point p,  
    double furthestDistance  
)
```

F#

```
member Nearest :  
    p : Point *  
    furthestDistance : float -> PriorityQueueRTree
```

Parameters

p

Type: [VelocityDBExtensions.Spatial.Point](#)

the point we are looking for

furthestDistance

Type: [System.Double](#)

The furthest distance away from the rectangle to search. Rectangles further than this will not be found.

Return Value

Type: [PriorityQueueRTree](#)

a PriorityQueue containing the found rectangles and their priorities (distances from point)

VelocityDB Class Library

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.NearestN Method

Finds the N nearest rectangles to the passed point. If multiple rectangles are equally near, they will all (but total limited to N) be returned.

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public PriorityQueueRTree NearestN(  
    Point p,  
    uint count,  
    double furthestDistance  
)
```

VB

```
Public Function NearestN (  
    p As Point,  
    count As UInteger,  
    furthestDistance As Double  
) As PriorityQueueRTree
```

C++

```
public:  
PriorityQueueRTree^ NearestN(  
    Point p,  
    unsigned int count,  
    double furthestDistance  
)
```

F#

```
member NearestN :  
    p : Point *  
    count : uint32 *  
    furthestDistance : float -> PriorityQueueRTree
```

Parameters

p

Type: [VelocityDBExtensions.Spatial.Point](#)

the point we are looking for

count

Type: [System.UInt32](#)

max number of rectangles to look for

furthestDistance

Type: [System.Double](#)

VelocityDB Class Library

The furthest distance away from the rectangle to search. Rectangles further than this will not be found.

Return Value

Type: [PriorityQueueRTree](#)

a PriorityQueue containing the found rectangles and their priorities (distances from point)

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.Remove Method

Removes a rectangle from the Rtree

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool Remove(  
    Rectangle r  
)
```

VB

```
Public Function Remove (  
    r As Rectangle  
) As Boolean
```

C++

```
public:  
bool Remove(  
    Rectangle r  
)
```

F#

```
member Remove :  
    r : Rectangle -> bool
```

Parameters

r

Type: [VelocityDBExtensions.Spatial.Rectangle](#)

the rectangle to delete

Return Value

Type: [Boolean](#)

true if rectangle deleted otherwise false

See Also



[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.RTree Fields

The [RTree](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|-----------------------------------------------------------|
|  | defaultMaxNodeEntries | default maximum number of rectangles in a leaf RTree node |
|  | defaultMinNodeEntries | default minimum number of rectangles in a leaf RTree node |

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.defaultMaxNodeEntries Field

default maximum number of rectangles in a leaf RTree node

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const int defaultMaxNodeEntries = 200
```

VB

```
Public Const defaultMaxNodeEntries As Integer = 200
```

C++

```
public:  
literal int defaultMaxNodeEntries = 200
```

F#

```
static val mutable defaultMaxNodeEntries: int
```

Field Value

Type: [Int32](#)

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

RTree.defaultMinNodeEntries Field

default minimum number of rectangles in a leaf RTree node

Namespace: [VelocityDBExtensions.Spatial](#)

Assembly: VelocityDBExtensions (in VelocityDBExtensions.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public const int defaultMinNodeEntries = 80
```

VB

```
Public Const defaultMinNodeEntries As Integer = 80
```

C++

```
public:  
literal int defaultMinNodeEntries = 80
```

F#

```
static val mutable defaultMinNodeEntries: int
```

Field Value

Type: [Int32](#)

See Also

[RTree Class](#)

[VelocityDBExtensions.Spatial Namespace](#)

VelocityDBExtensions2 Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions2"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-------------|
|  | AzureBlobSession | |
|  | AzureSession | |

AzureBlobSession Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AzureBlobSession"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

[VelocityDb.Session.SessionNoServer](#)

VelocityDBExtensions2.AzureBlobSession

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class AzureBlobSession : SessionNoServer
```

VB

```
Public Class AzureBlobSession
    Inherits SessionNoServer
```

C++


```
public ref class AzureBlobSession : public SessionNoServer
```

F#




```
type AzureBlobSession =
    class
        inherit SessionNoServer
    end
```






The **AzureBlobSession** type exposes the following members.

Constructors










| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
|  | AzureBlobSession | Initializes a new instance of the AzureBlobSession class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------|
|  | ContainsDatabase | (Overrides SessionBase.ContainsDatabase(DatabaseLocation, UInt32, String) .) |
|  | CreateDirectory | (Overrides SessionBase.CreateDirectory(String) .) |
|  | DatabaseStillExist | (Overrides SessionBase.DatabaseStillExist(Database) .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  DeleteFile | (Overrides SessionBase.DeleteFile(FileInfo).) |
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean).) |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean).) |
|  OpenAllDatabases | (Overrides SessionBase.OpenAllDatabases(Boolean).) |
|  WritePageBytes | (Overrides SessionBase.WritePageBytes(Byte[], Page, Dictionary(UInt16, Int64), UInt64, Stream, Byte[]).) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession Constructor

Initializes a new instance of the [AzureBlobSession](#) class

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public AzureBlobSession(
    string connectionString,
    string systemContainerName,
    int waitForLockMilliseconds = 5000,
    bool optimisticLocking = true,
    bool enableCache = true,
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes
)
```

VB

```
Public Sub New (
    connectionString As String,
    systemContainerName As String,
    Optional waitForLockMilliseconds As Integer = 5000,
    Optional optimisticLocking As Boolean = true,
    Optional enableCache As Boolean = true,
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes
)
```

C++

```
public:
    AzureBlobSession(
        String^ connectionString,
        String^ systemContainerName,
        int waitForLockMilliseconds = 5000,
        bool optimisticLocking = true,
        bool enableCache = true,
        CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes
    )
```

F#

```
new :
    connectionString : string *
    systemContainerName : string *
    ?waitForLockMilliseconds : int *
    ?optimisticLocking : bool *
    ?enableCache : bool *
    ?objectCachingDefaultPolicy : CacheEnum
(* Defaults:
    let _waitForLockMilliseconds = defaultArg waitForLockMilliseconds 5000
    let _optimisticLocking = defaultArg optimisticLocking true
    let _enableCache = defaultArg enableCache true
```



```
        let_objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
*)
-> AzureBlobSession
```

Parameters

connectionString

Type: [System.String](#)

[Missing <param name="connectionString"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

systemContaineName

Type: [System.String](#)

[Missing <param name="systemContaineName"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

[Missing <param name="waitForLockMilliseconds"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

optimisticLocking (Optional)

Type: [System.Boolean](#)

[Missing <param name="optimisticLocking"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

enableCache (Optional)

Type: [System.Boolean](#)

[Missing <param name="enableCache"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

[Missing <param name="objectCachingDefaultPolicy"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.#ctor(System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

See Also









[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)









AzureBlobSession.AzureBlobSession Methods

The [AzureBlobSession](#) type exposes the following members.


Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  ContainsDatabase | (Overrides SessionBase.ContainsDatabase(DatabaseLocation, UInt32, String) .) |
|  CreateDirectory | (Overrides SessionBase.CreateDirectory(String) .) |
|  DatabaseStillExist | (Overrides SessionBase.DatabaseStillExist(Database) .) |
|  DeleteFile | (Overrides SessionBase.DeleteFile(FileInfo) .) |
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) .) |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) .) |
|  OpenAllDatabases | (Overrides SessionBase.OpenAllDatabases(Boolean) .) |
|  WritePageBytes | (Overrides SessionBase.WritePageBytes(Byte[], Page, Dictionary(UInt16, Int64), UInt64, Stream, Byte[]) .) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv .) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport .) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport .) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport .) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv .) |
|  ImportJson(T) | (Defined by JsonImportExport .) |
|  MicrosoftSync | (Defined by Sync .) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync .) |

VelocityDB Class Library

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.ContainsDatabase Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool ContainsDatabase(
    DatabaseLocation location,
    uint dbNum,
    string extension = ".odb"
)
```

VB

```
Public Overrides Function ContainsDatabase (
    location As DatabaseLocation,
    dbNum As UInteger,
    Optional extension As String = ".odb"
) As Boolean
```

C++

```
public:
virtual bool ContainsDatabase(
    DatabaseLocation^ location,
    unsigned int dbNum,
    String^ extension = L".odb"
) override
```

F#

```
abstract ContainsDatabase :
    location : DatabaseLocation *
    dbNum : uint32 *
    ?extension : string
(* Defaults:
    let _extension = defaultArg extension ".odb"
*)
-> bool
override ContainsDatabase :
    location : DatabaseLocation *
    dbNum : uint32 *
    ?extension : string
(* Defaults:
    let _extension = defaultArg extension ".odb"
*)
-> bool
```

VelocityDB Class Library

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

[Missing <param name="location"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

dbNum

Type: [System.UInt32](#)

[Missing <param name="dbNum"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

extension (Optional)

Type: [System.String](#)

[Missing <param name="extension"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureBlobSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.CreateDirectory Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AzureBlobSession.CreateDirectory(System.String)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool CreateDirectory(  
    string path  
)
```

VB

```
Public Overrides Function CreateDirectory (  
    path As String  
) As Boolean
```

C++

```
public:  
virtual bool CreateDirectory(  
    String^ path  
) override
```

F#

```
abstract CreateDirectory :  
    path : string -> bool  
override CreateDirectory :  
    path : string -> bool
```

Parameters

path

Type: [System.String](#)

[Missing <param name="path"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.CreateDirectory(System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureBlobSession.CreateDirectory(System.String)"]

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.DatabaseStillExist Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.DatabaseStillExist(VelocityDb.Database)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool DatabaseStillExist(  
    Database db  
)
```

VB

```
Public Overrides Function DatabaseStillExist (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool DatabaseStillExist(  
    Database^ db  
) override
```

F#

```
abstract DatabaseStillExist :  
    db : Database -> bool  
override DatabaseStillExist :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.DatabaseStillExist(VelocityDb.Database)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.DatabaseStillExist(VelocityDb.Database)"]

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.DeleteFile Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AzureBlobSession.DeleteFile(System.IO.FileInfo)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void DeleteFile(  
    FileInfo fileInfo  
)
```

VB

```
Public Overrides Sub DeleteFile (  
    fileInfo As FileInfo  
)
```

C++

```
public:  
virtual void DeleteFile(  
    FileInfo^ fileInfo  
) override
```

F#

```
abstract DeleteFile :  
    fileInfo : FileInfo -> unit  
override DeleteFile :  
    fileInfo : FileInfo -> unit
```

Parameters

fileInfo

Type: [System.IO.FileInfo](#)

[Missing <param name="fileInfo"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.DeleteFile(System.IO.FileInfo)"]



See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.FileOpen Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  | FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) .) |
|  | FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) .) |

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.FileOpen Method (Database, FileAccess, String, FileMode, Boolean, Int32, Boolean)

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Stream FileOpen(  
    Database db,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileMode fileMode = FileMode.Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
)
```

VB

```
Public Overrides Function FileOpen (  
    db As Database,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    Optional fileMode As FileMode = FileMode.Open,  
    Optional exclusiveAccess As Boolean = false,  
    Optional waitOverride As Integer = -1,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    Database^ db,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileMode fileMode = FileMode::Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
) override
```

F#

```
abstract FileOpen :  
    db : Database *  
    fileAccess : FileAccess *  
    errorMessage : string byref *
```

```

        ?fileMode : FileMode *
        ?exclusiveAccess : bool *
        ?waitOverride : int *
        ?signalError : bool
    (* Defaults:
        let_fileMode = defaultArg fileMode FileMode.Open
        let_exclusiveAccess = defaultArg exclusiveAccess false
        let_waitOverride = defaultArg waitOverride -1
        let_signalError = defaultArg signalError true
    *)
    -> Stream
    override FileOpen :
        db : Database *
        fileAccess : FileAccess *
        errorMessage : string byref *
        ?fileMode : FileMode *
        ?exclusiveAccess : bool *
        ?waitOverride : int *
        ?signalError : bool
    (* Defaults:
        let_fileMode = defaultArg fileMode FileMode.Open
        let_exclusiveAccess = defaultArg exclusiveAccess false
        let_waitOverride = defaultArg waitOverride -1
        let_signalError = defaultArg signalError true
    *)
    -> Stream

```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

fileAccess

Type: [System.IO.FileAccess](#)

[Missing <param name="fileAccess"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

errorMessage

Type: [System.String](#)

[Missing <param name="errorMessage"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

fileMode (Optional)

Type: [System.IO.FileMode](#)

[Missing <param name="fileMode"/> documentation for "M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

VelocityDB Class Library

exclusiveAccess (Optional)

Type: [System.Boolean](#)

[Missing <param name="exclusiveAccess"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

waitOverride (Optional)

Type: [System.Int32](#)

[Missing <param name="waitOverride"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

signalError (Optional)

Type: [System.Boolean](#)

[Missing <param name="signalError"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

Return Value

Type: [Stream](#)

[Missing <returns> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

See Also

[AzureBlobSession Class](#)

[FileOpen Overload](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.FileOpen Method (FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean)

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Stream FileOpen(  
    FileInfo fileInfo,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
)
```

VB

```
Public Overrides Function FileOpen (  
    fileInfo As FileInfo,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    fileShare As FileShare,  
    fileMode As FileMode,  
    waitForLockMilliseconds As Integer,  
    useExternalStorage As Boolean,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    FileInfo^ fileInfo,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
) override
```

```

F#
abstract FileOpen :
    fileInfo : FileInfo *
    fileAccess : FileAccess *
    errorMessage : string byref *
    fileShare : FileShare *
    fileMode : FileMode *
    waitForLockMilliseconds : int *
    useExternalStorage : bool *
    ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream
override FileOpen :
    fileInfo : FileInfo *
    fileAccess : FileAccess *
    errorMessage : string byref *
    fileShare : FileShare *
    fileMode : FileMode *
    waitForLockMilliseconds : int *
    useExternalStorage : bool *
    ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream

```

Parameters

fileInfo

Type: [System.IO.FileInfo](#)

[Missing <param name="fileInfo"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

fileAccess

Type: [System.IO.FileAccess](#)

[Missing <param name="fileAccess"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

errorMessage

Type: [System.String](#)

[Missing <param name="errorMessage"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

fileShare

Type: [System.IO.FileShare](#)

[Missing <param name="fileShare"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

fileMode

Type: [System.IO.FileMode](#)

[Missing <param name="fileMode"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

waitForLockMilliseconds

Type: [System.Int32](#)

[Missing <param name="waitForLockMilliseconds"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

useExternalStorage

Type: [System.Boolean](#)

[Missing <param name="useExternalStorage"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

signalError (Optional)

Type: [System.Boolean](#)

[Missing <param name="signalError"/> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

Return Value

Type: [Stream](#)

[Missing <returns> documentation for
"M:VelocityDBExtensions2.AzureBlobSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

See Also

[AzureBlobSession Class](#)

[FileOpen Overload](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.OpenAllDatabases Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.OpenAllDatabases(System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
) override
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

[Missing <param name="update"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.OpenAllDatabases(System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [List\(Database\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.OpenAllDatabases(System.Boolean)"]

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureBlobSession.WritePageBytes Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void WritePageBytes (
    byte[] bytes,
    Page page,
    Dictionary<ushort, long> dbPageWrites,
    ulong transaction,
    Stream fStream,
    byte[] pageInfoBytes
)
```

VB

```
Public Overrides Sub WritePageBytes (
    bytes As Byte(),
    page As Page,
    dbPageWrites As Dictionary(Of UShort, Long),
    transaction As ULong,
    fStream As Stream,
    pageInfoBytes As Byte()
)
```

C++

```
public:
virtual void WritePageBytes (
    array<unsigned char>^ bytes,
    Page^ page,
    Dictionary<unsigned short, long long>^ dbPageWrites,
    unsigned long long transaction,
    Stream^ fStream,
    array<unsigned char>^ pageInfoBytes
) override
```

F#

```
abstract WritePageBytes :
    bytes : byte[] *
    page : Page *
    dbPageWrites : Dictionary<uint16, int64> *
    transaction : uint64 *
    fStream : Stream *
    pageInfoBytes : byte[] -> unit
```

```
override WritePageBytes :  
    bytes : byte[] *  
    page : Page *  
    dbPageWrites : Dictionary<uint16, int64> *  
    transaction : uint64 *  
    fStream : Stream *  
    pageInfoBytes : byte[] -> unit
```

Parameters

bytes

Type: [System.Byte\[\]](#)

[Missing <param name="bytes"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

page

Type: [VelocityDb.Page](#)

[Missing <param name="page"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

dbPageWrites

Type: [System.Collections.Generic.Dictionary{UInt16, Int64}](#)

[Missing <param name="dbPageWrites"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

transaction

Type: [System.UInt64](#)

[Missing <param name="transaction"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

fStream

Type: [System.IO.Stream](#)

[Missing <param name="fStream"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[])"

pageInfoBytes

Type: [System.Byte\[\]](#)

[Missing <param name="pageInfoBytes"/> documentation for

"M:VelocityDBExtensions2.AzureBlobSession.WritePageBytes(System.Byte[],VelocityDb.Page,System.

Collections.Generic.Dictionary{System.UInt16,System.Int64},System.UInt64,System.IO.Stream,System.Byte[]"

See Also

[AzureBlobSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AzureSession"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.Session.SessionBase](#)

[VelocityDb.Session.SessionNoServer](#)

VelocityDBExtensions2.AzureSession

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class AzureSession : SessionNoServer
```

VB

```
Public Class AzureSession
    Inherits SessionNoServer
```

C++


```
public ref class AzureSession : public SessionNoServer
```

F#




```
type AzureSession =
    class
        inherit SessionNoServer
    end
```





The **AzureSession** type exposes the following members.

Constructors










| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | AzureSession | Initializes a new instance of the AzureSession class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------|
|  | ContainsDatabase | (Overrides SessionBase.ContainsDatabase(DatabaseLocation, UInt32, String).) |
|  | CreateDirectory | (Overrides SessionBase.CreateDirectory(String).) |
|  | DatabaseStillExist | (Overrides SessionBase.DatabaseStillExist(Database).) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  DeleteFile | (Overrides SessionBase.DeleteFile(FileInfo).) |
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean).) |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean).) |
|  OpenAllDatabases | (Overrides SessionBase.OpenAllDatabases(Boolean).) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

See Also

[VelocityDBExtensions2 Namespace](#)

AzureSession Constructor

Initializes a new instance of the [AzureSession](#) class

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public AzureSession(  
    string connectionString,  
    string shareName,  
    string systemDir,  
    int waitForLockMilliseconds = 5000,  
    bool optimisticLocking = true,  
    bool enableCache = true,  
    CacheEnum objectCachingDefaultPolicy = CacheEnum.Yes  
)
```

VB

```
Public Sub New (  
    connectionString As String,  
    shareName As String,  
    systemDir As String,  
    Optional waitForLockMilliseconds As Integer = 5000,  
    Optional optimisticLocking As Boolean = true,  
    Optional enableCache As Boolean = true,  
    Optional objectCachingDefaultPolicy As CacheEnum = CacheEnum.Yes  
)
```

C++

```
public:  
AzureSession(  
    String^ connectionString,  
    String^ shareName,  
    String^ systemDir,  
    int waitForLockMilliseconds = 5000,  
    bool optimisticLocking = true,  
    bool enableCache = true,  
    CacheEnum objectCachingDefaultPolicy = CacheEnum::Yes  
)
```

F#

```
new :  
    connectionString : string *  
    shareName : string *  
    systemDir : string *  
    ?waitForLockMilliseconds : int *  
    ?optimisticLocking : bool *  
    ?enableCache : bool *  
    ?objectCachingDefaultPolicy : CacheEnum
```

```
(* Defaults:
    let _waitForLockMilliseconds = defaultArg waitForLockMilliseconds 5000
    let _optimisticLocking = defaultArg optimisticLocking true
    let _enableCache = defaultArg enableCache true
    let _objectCachingDefaultPolicy = defaultArg
objectCachingDefaultPolicy CacheEnum.Yes
*)
-> AzureSession
```

Parameters

connectionString

Type: [System.String](#)

[Missing <param name="connectionString"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

shareName

Type: [System.String](#)

[Missing <param name="shareName"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

systemDir

Type: [System.String](#)

[Missing <param name="systemDir"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

waitForLockMilliseconds (Optional)

Type: [System.Int32](#)

[Missing <param name="waitForLockMilliseconds"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

optimisticLocking (Optional)

Type: [System.Boolean](#)

[Missing <param name="optimisticLocking"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

enableCache (Optional)

Type: [System.Boolean](#)

[Missing <param name="enableCache"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

objectCachingDefaultPolicy (Optional)

Type: [VelocityDb.CacheEnum](#)

[Missing <param name="objectCachingDefaultPolicy"/> documentation for "M:VelocityDBExtensions2.AzureSession.#ctor(System.String,System.String,System.String,System.Int32,System.Boolean,System.Boolean,VelocityDb.CacheEnum)"]

See Also








[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)










AzureSession.AzureSession Methods

The [AzureSession](#) type exposes the following members.

Methods

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  ContainsDatabase | (Overrides SessionBase.ContainsDatabase(DatabaseLocation, UInt32, String).) |
|  CreateDirectory | (Overrides SessionBase.CreateDirectory(String).) |
|  DatabaseStillExist | (Overrides SessionBase.DatabaseStillExist(Database).) |
|  DeleteFile | (Overrides SessionBase.DeleteFile(FileInfo).) |
|  FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean).) |
|  FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean).) |
|  OpenAllDatabases | (Overrides SessionBase.OpenAllDatabases(Boolean).) |

Extension Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ExportToCSV | Export all persistent objects to .csv files, one file for each Type and version of Type. This is preview release, format may change. ImportFromCSV can be used to recreate your data. Note that Microsoft Excel can't handle many of these CSV files due to a field value limitation (at about 33000 chars) Notepad++ is one application that can read these files. Some fields like array data are encoded http://msdn.microsoft.com/en-us/library/dhx0d524(v=vs.110).aspx (Defined by ImportExportCsv.) |
|  ExportToJson(T)(UInt64) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Oid) | Overloaded. (Defined by JsonImportExport.) |
|  ExportToJson(T)(Boolean, Boolean) | Overloaded. (Defined by JsonImportExport.) |
|  ImportFromCSV | Restores database files, pages and objects from a .csv file data created with ExportToCSV (Defined by ImportExportCsv.) |
|  ImportJson(T) | (Defined by JsonImportExport.) |
|  MicrosoftSync | (Defined by Sync.) |
|  SyncWith(SessionBase) | Overloaded. (Defined by Sync.) |
|  SyncWith(SessionBase, Func(SessionBase, UInt64, Change, Boolean)) | Overloaded. (Defined by Sync.) |

VelocityDB Class Library

See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.ContainsDatabase Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool ContainsDatabase(
    DatabaseLocation location,
    uint dbNum,
    string extension = ".odb"
)
```

VB

```
Public Overrides Function ContainsDatabase (
    location As DatabaseLocation,
    dbNum As UInteger,
    Optional extension As String = ".odb"
) As Boolean
```

C++

```
public:
virtual bool ContainsDatabase(
    DatabaseLocation^ location,
    unsigned int dbNum,
    String^ extension = L".odb"
) override
```

F#

```
abstract ContainsDatabase :
    location : DatabaseLocation *
    dbNum : uint32 *
    ?extension : string
(* Defaults:
    let _extension = defaultArg extension ".odb"
*)
-> bool
override ContainsDatabase :
    location : DatabaseLocation *
    dbNum : uint32 *
    ?extension : string
(* Defaults:
    let _extension = defaultArg extension ".odb"
*)
-> bool
```

Parameters

location

Type: [VelocityDb.DatabaseLocation](#)

[Missing <param name="location"/> documentation for "M:VelocityDBExtensions2.AzureSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

dbNum

Type: [System.UInt32](#)

[Missing <param name="dbNum"/> documentation for "M:VelocityDBExtensions2.AzureSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

extension (Optional)

Type: [System.String](#)

[Missing <param name="extension"/> documentation for "M:VelocityDBExtensions2.AzureSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureSession.ContainsDatabase(VelocityDb.DatabaseLocation,System.UInt32,System.String)"]

See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.CreateDirectory Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AzureSession.CreateDirectory(System.String)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool CreateDirectory(  
    string path  
)
```

VB

```
Public Overrides Function CreateDirectory (  
    path As String  
) As Boolean
```

C++

```
public:  
virtual bool CreateDirectory(  
    String^ path  
) override
```

F#

```
abstract CreateDirectory :  
    path : string -> bool  
override CreateDirectory :  
    path : string -> bool
```

Parameters

path

Type: [System.String](#)

[Missing <param name="path"/> documentation for "M:VelocityDBExtensions2.AzureSession.CreateDirectory(System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureSession.CreateDirectory(System.String)"]

See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.DatabaseStillExist Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AzureSession.DatabaseStillExist(VelocityDb.Database)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override bool DatabaseStillExist(  
    Database db  
)
```

VB

```
Public Overrides Function DatabaseStillExist (  
    db As Database  
) As Boolean
```

C++

```
public:  
virtual bool DatabaseStillExist(  
    Database^ db  
) override
```

F#

```
abstract DatabaseStillExist :  
    db : Database -> bool  
override DatabaseStillExist :  
    db : Database -> bool
```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDBExtensions2.AzureSession.DatabaseStillExist(VelocityDb.Database)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureSession.DatabaseStillExist(VelocityDb.Database)"]

See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.DeleteFile Method

[Missing <summary> documentation for
"M:VelocityDBExtensions2.AzureSession.DeleteFile(System.IO.FileInfo)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override void DeleteFile(  
    FileInfo fileInfo  
)
```

VB

```
Public Overrides Sub DeleteFile (  
    fileInfo As FileInfo  
)
```

C++

```
public:  
virtual void DeleteFile(  
    FileInfo^ fileInfo  
) override
```

F#

```
abstract DeleteFile :  
    fileInfo : FileInfo -> unit  
override DeleteFile :  
    fileInfo : FileInfo -> unit
```

Parameters

fileInfo

Type: [System.IO.FileInfo](#)

[Missing <param name="fileInfo"/> documentation for
"M:VelocityDBExtensions2.AzureSession.DeleteFile(System.IO.FileInfo)"]



See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.FileOpen Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|  | FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) | (Overrides SessionBase.FileOpen(Database, FileAccess, String, FileMode, Boolean, Int32, Boolean) .) |
|  | FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) | (Overrides SessionBase.FileOpen(FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean) .) |

See Also

[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.FileOpen Method (Database, FileAccess, String, FileMode, Boolean, Int32, Boolean)

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Stream FileOpen(  
    Database db,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileMode fileMode = FileMode.Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
)
```

VB

```
Public Overrides Function FileOpen (  
    db As Database,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    Optional fileMode As FileMode = FileMode.Open,  
    Optional exclusiveAccess As Boolean = false,  
    Optional waitOverride As Integer = -1,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    Database^ db,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileMode fileMode = FileMode::Open,  
    bool exclusiveAccess = false,  
    int waitOverride = -1,  
    bool signalError = true  
) override
```

F#

```
abstract FileOpen :  
    db : Database *  
    fileAccess : FileAccess *  
    errorMessage : string byref *
```

```

        ?fileMode : FileMode *
        ?exclusiveAccess : bool *
        ?waitOverride : int *
        ?signalError : bool
    (* Defaults:
        let_fileMode = defaultArg fileMode FileMode.Open
        let_exclusiveAccess = defaultArg exclusiveAccess false
        let_waitOverride = defaultArg waitOverride -1
        let_signalError = defaultArg signalError true
    *)
    -> Stream
    override FileOpen :
        db : Database *
        fileAccess : FileAccess *
        errorMessage : string byref *
        ?fileMode : FileMode *
        ?exclusiveAccess : bool *
        ?waitOverride : int *
        ?signalError : bool
    (* Defaults:
        let_fileMode = defaultArg fileMode FileMode.Open
        let_exclusiveAccess = defaultArg exclusiveAccess false
        let_waitOverride = defaultArg waitOverride -1
        let_signalError = defaultArg signalError true
    *)
    -> Stream

```

Parameters

db

Type: [VelocityDb.Database](#)

[Missing <param name="db"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

fileAccess

Type: [System.IO.FileAccess](#)

[Missing <param name="fileAccess"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

errorMessage

Type: [System.String](#)

[Missing <param name="errorMessage"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

fileMode (Optional)

Type: [System.IO.FileMode](#)

[Missing <param name="fileMode"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

VelocityDB Class Library

exclusiveAccess (Optional)

Type: [System.Boolean](#)

[Missing <param name="exclusiveAccess"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

waitOverride (Optional)

Type: [System.Int32](#)

[Missing <param name="waitOverride"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

signalError (Optional)

Type: [System.Boolean](#)

[Missing <param name="signalError"/> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

Return Value

Type: [Stream](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AzureSession.FileOpen(VelocityDb.Database,System.IO.FileAccess,System.String@,System.IO.FileMode,System.Boolean,System.Int32,System.Boolean)"]

See Also

[AzureSession Class](#)

[FileOpen Overload](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.FileOpen Method (FileInfo, FileAccess, String, FileShare, FileMode, Int32, Boolean, Boolean)

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override Stream FileOpen(  
    FileInfo fileInfo,  
    FileAccess fileAccess,  
    ref string errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
)
```

VB

```
Public Overrides Function FileOpen (  
    fileInfo As FileInfo,  
    fileAccess As FileAccess,  
    ByRef errorMessage As String,  
    fileShare As FileShare,  
    fileMode As FileMode,  
    waitForLockMilliseconds As Integer,  
    useExternalStorage As Boolean,  
    Optional signalError As Boolean = true  
) As Stream
```

C++

```
public:  
virtual Stream^ FileOpen(  
    FileInfo^ fileInfo,  
    FileAccess fileAccess,  
    String^% errorMessage,  
    FileShare fileShare,  
    FileMode fileMode,  
    int waitForLockMilliseconds,  
    bool useExternalStorage,  
    bool signalError = true  
) override
```

F#

```

abstract FileOpen :
    fileInfo : FileInfo *
    fileAccess : FileAccess *
    errorMessage : string byref *
    fileShare : FileShare *
    fileMode : FileMode *
    waitForLockMilliseconds : int *
    useExternalStorage : bool *
    ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream
override FileOpen :
    fileInfo : FileInfo *
    fileAccess : FileAccess *
    errorMessage : string byref *
    fileShare : FileShare *
    fileMode : FileMode *
    waitForLockMilliseconds : int *
    useExternalStorage : bool *
    ?signalError : bool
(* Defaults:
    let _signalError = defaultArg signalError true
*)
-> Stream

```

*Parameters**fileInfo*Type: [System.IO.FileInfo](#)**[Missing <param name="fileInfo"/> documentation for****"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]***fileAccess*Type: [System.IO.FileAccess](#)**[Missing <param name="fileAccess"/> documentation for****"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]***errorMessage*Type: [System.String](#)**[Missing <param name="errorMessage"/> documentation for****"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]***fileShare*Type: [System.IO.FileShare](#)

[Missing <param name="fileShare"/> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

fileMode

Type: [System.IO.FileMode](#)

[Missing <param name="fileMode"/> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

waitForLockMilliseconds

Type: [System.Int32](#)

[Missing <param name="waitForLockMilliseconds"/> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

useExternalStorage

Type: [System.Boolean](#)

[Missing <param name="useExternalStorage"/> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

signalError (Optional)

Type: [System.Boolean](#)

[Missing <param name="signalError"/> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

Return Value

Type: [Stream](#)

[Missing <returns> documentation for
"M:VelocityDBExtensions2.AzureSession.FileOpen(System.IO.FileInfo,System.IO.FileAccess,System.String@,System.IO.FileShare,System.IO.FileMode,System.Int32,System.Boolean,System.Boolean)"]

See Also

[AzureSession Class](#)

[FileOpen Overload](#)

[VelocityDBExtensions2 Namespace](#)

AzureSession.OpenAllDatabases Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AzureSession.OpenAllDatabases(System.Boolean)"]

Namespace: [VelocityDBExtensions2](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override List<Database> OpenAllDatabases (  
    bool update = false  
)
```

VB

```
Public Overrides Function OpenAllDatabases (  
    Optional update As Boolean = false  
) As List(Of Database)
```

C++

```
public:  
virtual List<Database^>^ OpenAllDatabases (  
    bool update = false  
) override
```

F#

```
abstract OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>  
override OpenAllDatabases :  
    ?update : bool  
(* Defaults:  
    let_update = defaultArg update false  
*)  
-> List<Database>
```

Parameters

update (Optional)

Type: [System.Boolean](#)

[Missing <param name="update"/> documentation for

"M:VelocityDBExtensions2.AzureSession.OpenAllDatabases(System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [List\(Database\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AzureSession.OpenAllDatabases(System.Boolean)"]

See Also








[AzureSession Class](#)

[VelocityDBExtensions2 Namespace](#)

VelocityDBExtensions2.AspNet.Identity Namespace

[Missing <summary> documentation for "N:VelocityDBExtensions2.AspNet.Identity"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | AspNetIdentity | |
|  | IdentityRole | |
|  | IdentityUser | |
|  | KeyFormats | |
|  | RoleStore(T) | |
|  | UserLoginInfoAdapter | |
|  | UserStore(T) | |

AspNetIdentity Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions2.AspNet.Identity.AspNetIdentity

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class AspNetIdentity : OptimizedPersistable
```

VB

```
Public Class AspNetIdentity
    Inherits OptimizedPersistable
```

C++


```
public ref class AspNetIdentity : public OptimizedPersistable
```

F#






```
type AspNetIdentity =
    class
        inherit OptimizedPersistable
    end
```

The **AspNetIdentity** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | AspNetIdentity | Initializes a new instance of the AspNetIdentity class |

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | AdapterMap | |
|  | EmailTold | |
|  | Name | |
|  | RoleSet | |
|  | UserNameTold | |

| | | |
|-----------------------------------------------------------------------------------|-------------------------|--|
|  | UserSet | |
|-----------------------------------------------------------------------------------|-------------------------|--|

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity Constructor

Initializes a new instance of the [AspNetIdentity](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public AspNetIdentity(  
    SessionBase session  
)
```

VB

```
Public Sub New (  
    session As SessionBase  
)
```

C++

```
public:  
AspNetIdentity(  
    SessionBase^ session  
)
```

F#

```
new :  
    session : SessionBase -> AspNetIdentity
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.#ctor(VelocityDb.Session.SessionBase)"]

See Also







[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.AspNetIdentity Properties

The [AspNetIdentity](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | AdapterManager | |
|  | EmailTold | |
|  | Name | |
|  | RoleSet | |
|  | UserNameTold | |
|  | UserSet | |

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.AdapterMap Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.AdapterMap"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BTreeMap<string, UserLoginInfoAdapter> AdapterMap { get; }
```

VB

```
Public ReadOnly Property AdapterMap As BTreeMap(Of String,  
UserLoginInfoAdapter)  
    Get
```

C++

```
public:  
property BTreeMap<String^, UserLoginInfoAdapter^>^ AdapterMap {  
    BTreeMap<String^, UserLoginInfoAdapter^>^ get ();  
}
```

F#

```
member AdapterMap : BTreeMap<string, UserLoginInfoAdapter> with get
```

Property Value

Type: [BTreeMap\(String, UserLoginInfoAdapter\)](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.EmailTold Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.EmailTold"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BTreeMap<string, ulong> EmailToId { get; }
```

VB

```
Public ReadOnly Property EmailToId As BTreeMap(Of String, ULong)  
    Get
```

C++

```
public:  
property BTreeMap<String^, unsigned long long>^ EmailToId {  
    BTreeMap<String^, unsigned long long>^ get ();  
}
```

F#

```
member EmailToId : BTreeMap<string, uint64> with get
```

Property Value

Type: [BTreeMap\(String, UInt64\)](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.Name Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Name"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string Name { get; set; }
```

VB

```
Public Property Name As String  
    Get  
    Set
```

C++

```
public:  
property String^ Name {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member Name : string with get, set
```

Property Value

Type: [String](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.RoleSet Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.RoleSet"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BTreeSet<IdentityRole> RoleSet { get; }
```

VB

```
Public ReadOnly Property RoleSet As BTreeSet(Of IdentityRole)  
    Get
```

C++

```
public:  
property BTreeSet<IdentityRole^>^ RoleSet {  
    BTreeSet<IdentityRole^>^ get ();  
}
```

F#

```
member RoleSet : BTreeSet<IdentityRole> with get
```

Property Value

Type: [BTreeSet<IdentityRole>](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.UserNameTold Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.UserNameTold"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BTreeMap<string, ulong> UserNameToId { get; }
```

VB

```
Public ReadOnly Property UserNameToId As BTreeMap(Of String, ULong)  
    Get
```

C++

```
public:  
property BTreeMap<String^, unsigned long long>^ UserNameToId {  
    BTreeMap<String^, unsigned long long>^ get ();  
}
```

F#

```
member UserNameToId : BTreeMap<string, uint64> with get
```

Property Value

Type: [BTreeMap\(String, UInt64\)](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.UserSet Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.UserSet"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public BTreeSet<IdentityUser> UserSet { get; }
```

VB

```
Public ReadOnly Property UserSet As BTreeSet(Of IdentityUser)  
    Get
```

C++

```
public:  
property BTreeSet<IdentityUser^>^ UserSet {  
    BTreeSet<IdentityUser^>^ get ();  
}
```

F#

```
member UserSet : BTreeSet<IdentityUser> with get
```

Property Value

Type: [BTreeSet<IdentityUser>](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)



AspNetIdentity.AspNetIdentity Methods

The [AspNetIdentity](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  | Persist | (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

AspNetIdentity.Persist Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _persistRefs = defaultArg persistRefs true
    let _disableFlush = defaultArg disableFlush false
    let _toPersist = defaultArg toPersist null
```

```
*)
-> uint64
override Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

[Missing <param name="place"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

persistRefs (Optional)

Type: [System.Boolean](#)

[Missing <param name="persistRefs"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

disableFlush (Optional)

Type: [System.Boolean](#)

[Missing <param name="disableFlush"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.IOptimizedPersistable})"]

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

[Missing <param name="toPersist"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.

Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.I OptimizedPersistable}"]

Return Value

Type: [UInt64](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.AspNetIdentity.Persist(VelocityDb.Placement,VelocityDb.Session.SessionBase,System.Boolean,System.Boolean,System.Collections.Generic.Queue{VelocityDb.I OptimizedPersistable}"]

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue\(IOptimizedPersistable\)\)](#)

See Also

[AspNetIdentity Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.IdentityRole"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions2.AspNet.Identity.IdentityRole

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class IdentityRole : OptimizedPersistable,
    IRole<ulong>
```

VB

```
Public Class IdentityRole
    Inherits OptimizedPersistable
    Implements IRole(Of ULong)
```

C++



```
public ref class IdentityRole : public OptimizedPersistable,
    IRole<unsigned long long>
```

F#

```
type IdentityRole =
    class
        inherit OptimizedPersistable
        interface IRole<uint64>
    end
```

The **IdentityRole** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------|
|  | IdentityRole() | Initializes a new instance of the IdentityRole class |
|  | IdentityRole(String) | Initializes a new instance of the IdentityRole class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |

Extension Methods



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------|
|  | IdentityRole() | Initializes a new instance of the IdentityRole class |
|  | IdentityRole(String) | Initializes a new instance of the IdentityRole class |

See Also

[IdentityRole Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole Constructor

Initializes a new instance of the [IdentityRole](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IdentityRole ()
```

VB

```
Public Sub New
```

C++

```
public:  
IdentityRole ()
```

F#

```
new : unit -> IdentityRole
```

See Also

[IdentityRole Class](#)

[IdentityRole Overload](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole Constructor (String)

Initializes a new instance of the [IdentityRole](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IdentityRole(  
    string name  
)
```

VB

```
Public Sub New (  
    name As String  
)
```

C++

```
public:  
IdentityRole(  
    String^ name  
)
```

F#

```
new :  
    name : string -> IdentityRole
```

Parameters

name

Type: [System.String](#)

[Missing <param name="name"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.IdentityRole.#ctor(System.String)"]

See Also

[IdentityRole Class](#)


[IdentityRole Overload](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole.IdentityRole Properties

The [IdentityRole](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |

See Also

[IdentityRole Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole.Name Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityRole.Name"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string Name { get; set; }
```

VB

```
Public Property Name As String  
    Get  
    Set
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

F#

```
abstract Name : string with get, set  
override Name : string with get, set
```

Property Value

Type: [String](#)

Implements

[IRole.Name\(\)](#)

See Also



[IdentityRole Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityRole.IdentityRole Methods

The [IdentityRole](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[IdentityRole Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.IdentityUser"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions2.AspNet.Identity.IdentityUser

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class IdentityUser : OptimizedPersistable,
    IUser<ulong>
```

VB

```
Public Class IdentityUser
    Inherits OptimizedPersistable
    Implements IUser(Of ULong)
```

C++



```
public ref class IdentityUser : public OptimizedPersistable,
    IUser<unsigned long long>
```

F#

```
type IdentityUser =
    class
        inherit OptimizedPersistable
        interface IUser<uint64>
    end
```

The **IdentityUser** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------|
|  | IdentityUser() | Initializes a new instance of the IdentityUser class |
|  | IdentityUser(String) | Initializes a new instance of the IdentityUser class |



Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | AccessFailedCount | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------|--|
|  | Claims | |
|  | Email | |
|  | EmailConfirmed | |
|  | LockoutEnabled | |
|  | LockoutEndDateUtc | |
|  | PasswordHash | |
|  | PhoneNumber | |
|  | PhoneNumberConfirmed | |
|  | Roles | |
|  | SecurityStamp | |
|  | TwoFactorEnabled | |
|  | UserLoginIds | |
|  | UserName | |

Extension Methods



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|----------------------------------------------------------------------|
|  | IdentityUser() | Initializes a new instance of the IdentityUser class |
|  | IdentityUser(String) | Initializes a new instance of the IdentityUser class |

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser Constructor

Initializes a new instance of the [IdentityUser](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IdentityUser ()
```

VB

```
Public Sub New
```

C++

```
public:  
IdentityUser ()
```

F#

```
new : unit -> IdentityUser
```

See Also

[IdentityUser Class](#)

[IdentityUser Overload](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser Constructor (String)

Initializes a new instance of the [IdentityUser](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IdentityUser(  
    string username  
)
```

VB

```
Public Sub New (  
    username As String  
)
```

C++

```
public:  
IdentityUser(  
    String^ username  
)
```

F#

```
new :  
    username : string -> IdentityUser
```

Parameters

username

Type: [System.String](#)

[Missing <param name="username"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.IdentityUser.#ctor(System.String)"]

See Also

[IdentityUser Class](#)















[IdentityUser Overload](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.IdentityUser Properties

The [IdentityUser](#) type exposes the following members.

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | AccessFailedCount | |
|  | Claims | |
|  | Email | |
|  | EmailConfirmed | |
|  | LockoutEnabled | |
|  | LockoutEndDateUtc | |
|  | PasswordHash | |
|  | PhoneNumber | |
|  | PhoneNumberConfirmed | |
|  | Roles | |
|  | SecurityStamp | |
|  | TwoFactorEnabled | |
|  | UserLoginIds | |
|  | UserName | |

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.AccessFailedCount Property

[Missing <summary> documentation for

"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.AccessFailedCount"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public int AccessFailedCount { get; set; }
```

VB

```
Public Property AccessFailedCount As Integer  
    Get  
    Set
```

C++

```
public:  
property int AccessFailedCount {  
    int get ();  
    void set (int value);  
}
```

F#

```
member AccessFailedCount : int with get, set
```

Property Value

Type: [Int32](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.Claims Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.Claims"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public ICollection<Claim> Claims { get; set; }
```

VB

```
Public Property Claims As ICollection(Of Claim)  
    Get  
    Set
```

C++

```
public:  
property ICollection<Claim^> Claims {  
    ICollection<Claim^> get ();  
    void set (ICollection<Claim^> value);  
}
```

F#

```
member Claims : ICollection<Claim> with get, set
```

Property Value

Type: [ICollection\(Claim\)](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.Email Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.Email"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string Email { get; set; }
```

VB

```
Public Property Email As String  
    Get  
    Set
```

C++

```
public:  
property String^ Email {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member Email : string with get, set
```

Property Value

Type: [String](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.EmailConfirmed Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.EmailConfirmed"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool EmailConfirmed { get; set; }
```

VB

```
Public Property EmailConfirmed As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool EmailConfirmed {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member EmailConfirmed : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.LockoutEnabled Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.LockoutEnabled"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool LockoutEnabled { get; set; }
```

VB

```
Public Property LockoutEnabled As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool LockoutEnabled {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member LockoutEnabled : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.LockoutEndDateUtc Property

[Missing <summary> documentation for

"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.LockoutEndDateUtc"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public DateTime? LockoutEndDateUtc { get; set; }
```

VB

```
Public Property LockoutEndDateUtc As DateTime?  
    Get  
    Set
```

C++

```
public:  
property Nullable<DateTime> LockoutEndDateUtc {  
    Nullable<DateTime> get ();  
    void set (Nullable<DateTime> value);  
}
```

F#

```
member LockoutEndDateUtc : Nullable<DateTime> with get, set
```

Property Value

Type: [Nullable\(DateTime\)](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.PasswordHash Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.PasswordHash"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string PasswordHash { get; set; }
```

VB

```
Public Property PasswordHash As String  
    Get  
    Set
```

C++

```
public:  
property String^ PasswordHash {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member PasswordHash : string with get, set
```

Property Value

Type: [String](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.PhoneNumber Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.PhoneNumber"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string PhoneNumber { get; set; }
```

VB

```
Public Property PhoneNumber As String  
    Get  
    Set
```

C++

```
public:  
property String^ PhoneNumber {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member PhoneNumber : string with get, set
```

Property Value

Type: [String](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.PhoneNumberConfirmed Property

[Missing <summary> documentation for

"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.PhoneNumberConfirmed"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool PhoneNumberConfirmed { get; set; }
```

VB

```
Public Property PhoneNumberConfirmed As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool PhoneNumberConfirmed {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member PhoneNumberConfirmed : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.Roles Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.Roles"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public ICollection<IdentityRole> Roles { get; set; }
```

VB

```
Public Property Roles As ICollection(Of IdentityRole)  
    Get  
    Set
```

C++

```
public:  
property ICollection<IdentityRole^> Roles {  
    ICollection<IdentityRole^> get ();  
    void set (ICollection<IdentityRole^> value);  
}
```

F#

```
member Roles : ICollection<IdentityRole> with get, set
```

Property Value

Type: [ICollection\(IdentityRole\)](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.SecurityStamp Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.SecurityStamp"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string SecurityStamp { get; set; }
```

VB

```
Public Property SecurityStamp As String  
    Get  
    Set
```

C++

```
public:  
property String^ SecurityStamp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member SecurityStamp : string with get, set
```

Property Value

Type: [String](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.TwoFactorEnabled Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.TwoFactorEnabled"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public bool TwoFactorEnabled { get; set; }
```

VB

```
Public Property TwoFactorEnabled As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool TwoFactorEnabled {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member TwoFactorEnabled : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.UserLoginIds Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.UserLoginIds"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IList<UserLoginInfo> UserLoginIds { get; set; }
```

VB

```
Public Property UserLoginIds As IList(Of UserLoginInfo)  
    Get  
    Set
```

C++

```
public:  
property IList<UserLoginInfo^>^ UserLoginIds {  
    IList<UserLoginInfo^>^ get ();  
    void set (IList<UserLoginInfo^>^ value);  
}
```

F#

```
member UserLoginIds : IList<UserLoginInfo> with get, set
```

Property Value

Type: [IList<UserLoginInfo>](#)

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.UserName Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.IdentityUser.UserName"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public string UserName { get; set; }
```

VB

```
Public Property UserName As String  
    Get  
    Set
```

C++

```
public:  
virtual property String^ UserName {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

F#

```
abstract UserName : string with get, set  
override UserName : string with get, set
```

Property Value

Type: [String](#)

Implements

IUser.UserName()

See Also



[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

IdentityUser.IdentityUser Methods

The [IdentityUser](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[IdentityUser Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.KeyFormats"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions2.AspNet.Identity.KeyFormats

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static class KeyFormats
```

VB

```
Public NotInheritable Class KeyFormats
```

C++


```
public ref class KeyFormats abstract sealed
```

F#



```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type KeyFormats = class end
```

The **KeyFormats** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-------------|
|  | GetKey | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | LoginFormat | |
|  | UserFormat | |


See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats.KeyFormats Methods

The [KeyFormats](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | GetKey | |

See Also

[KeyFormats Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats.GetKey Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.KeyFormats.GetKey(System.Object,System.Object)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string GetKey(  
    Object loginKey,  
    Object userKey  
)
```

VB

```
Public Shared Function GetKey (  
    loginKey As Object,  
    userKey As Object  
) As String
```

C++

```
public:  
static String^ GetKey(  
    Object^ loginKey,  
    Object^ userKey  
)
```

F#

```
static member GetKey :  
    loginKey : Object *  
    userKey : Object -> string
```

Parameters

loginKey

Type: [System.Object](#)

[Missing <param name="loginKey"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.KeyFormats.GetKey(System.Object,System.Object)"]

userKey

Type: [System.Object](#)

[Missing <param name="userKey"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.KeyFormats.GetKey(System.Object,System.Object)"]

VelocityDB Class Library

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.KeyFormats.GetKey(System.Object,System.Object)"]

See Also

[KeyFormats Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats.KeyFormats Fields

The [KeyFormats](#) type exposes the following members.

Fields

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-------------|
|   | LoginFormat | |
|   | UserFormat | |

See Also

[KeyFormats Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats.LoginFormat Field

[Missing <summary> documentation for "F:VelocityDBExtensions2.AspNet.Identity.KeyFormats.LoginFormat"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string LoginFormat
```

VB

```
Public Shared LoginFormat As String
```

C++

```
public:  
static String^ LoginFormat
```

F#

```
static val mutable LoginFormat: string
```

Field Value

Type: [String](#)

See Also

[KeyFormats Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

KeyFormats.UserFormat Field

[Missing <summary> documentation for "F:VelocityDBExtensions2.AspNet.Identity.KeyFormats.UserFormat"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public static string UserFormat
```

VB

```
Public Shared UserFormat As String
```

C++

```
public:  
static String^ UserFormat
```

F#

```
static val mutable UserFormat: string
```

Field Value

Type: [String](#)

See Also

[KeyFormats Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T) Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.RoleStore`1"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions2.AspNet.Identity.RoleStore(T)

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class RoleStore<T> : IQueryableRoleStore<T, ulong>,
    IRoleStore<T, ulong>, IDisposable
where T : IdentityRole
```

VB

```
Public Class RoleStore(Of T As IdentityRole)
    Implements IQueryableRoleStore(Of T, ULong), IRoleStore(Of T, ULong),
    IDisposable
```

C++

```
generic<typename T>
where T : IdentityRole
public ref class RoleStore : IQueryableRoleStore<T, unsigned long long>,
    IRoleStore<T, unsigned long long>, IDisposable
```

F#

```
type RoleStore<'T when 'T : IdentityRole> =
    class
        interface IQueryableRoleStore<'T, uint64>
        interface IRoleStore<'T, uint64>
        interface IDisposable
    end
```


Type Parameters

T


[Missing <typeparam name="T"/> documentation for "T:VelocityDBExtensions2.AspNet.Identity.RoleStore`1"]

The RoleStore(T) type exposes the following members.







Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|------------------------------------------------------|
|  | RoleStore(T) | Initializes a new instance of the RoleStore(T) class |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Roles | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-------------------------------------------------|
|  | CreateAsync | |
|  | DeleteAsync | |
|  | Dispose | Releases all resources used by the RoleStore(T) |
|  | FindByIdAsync | |
|  | FindByNameAsync | |
|  | UpdateAsync | |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T) Constructor

Initializes a new instance of the [RoleStore\(T\)](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public RoleStore(  
    AspNetIdentity identity  
)
```

VB

```
Public Sub New (  
    identity As AspNetIdentity  
)
```

C++

```
public:  
RoleStore(  
    AspNetIdentity^ identity  
)
```

F#

```
new :  
    identity : AspNetIdentity -> RoleStore
```

Parameters

identity

Type: [VelocityDBExtensions2.AspNet.Identity.AspNetIdentity](#)

[Missing <param name="identity"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.#ctor(VelocityDBExtensions2.AspNet.Identity.AspNetIdentity)"]

See Also


[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).RoleStore(T) Properties

The [RoleStore\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Roles | |

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).Roles Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.Roles"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IQueryable<T> Roles { get; }
```

VB

```
Public ReadOnly Property Roles As IQueryable(Of T)  
    Get
```

C++

```
public:  
virtual property IQueryable<T>^ Roles {  
    IQueryable<T>^ get () sealed;  
}
```

F#

```
abstract Roles : IQueryable<'T> with get  
override Roles : IQueryable<'T> with get
```

Property Value

Type: [IQueryable\(T\)](#)

Implements

[IQueryableRoleStore.Roles\(\)](#)

See Also







[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).RoleStore(T) Methods

The [RoleStore\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-------------|
|  | CreateAsync | |
|  | DeleteAsync | |
|  | Dispose | |
|  | FindByIdAsync | |
|  | FindByNameAsync | |
|  | UpdateAsync | |

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).CreateAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.CreateAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task CreateAsync(  
    T role  
)
```

VB

```
Public Function CreateAsync (  
    role As T  
) As Task
```

C++

```
public:  
virtual Task^ CreateAsync(  
    T role  
) sealed
```

F#

```
abstract CreateAsync :  
    role : 'T -> Task  
override CreateAsync :  
    role : 'T -> Task
```

Parameters

role

Type: *T*

[Missing <param name="role"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.CreateAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.CreateAsync(`0)"]

Implements

IRoleStore.CreateAsync(UTP)

VelocityDB Class Library

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).DeleteAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.DeleteAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task DeleteAsync(  
    T role  
)
```

VB

```
Public Function DeleteAsync (  
    role As T  
) As Task
```

C++

```
public:  
virtual Task^ DeleteAsync(  
    T role  
) sealed
```

F#

```
abstract DeleteAsync :  
    role : 'T -> Task  
override DeleteAsync :  
    role : 'T -> Task
```

Parameters

role

Type: *T*

[Missing <param name="role"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.DeleteAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.DeleteAsync(`0)"]

Implements

IRoleStore.DeleteAsync(UTP)

VelocityDB Class Library

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).Dispose Method

Releases all resources used by the [RoleStore\(T\)](#)

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).FindByIdAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByIdAsync(System.UInt64)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindByIdAsync(  
    ulong roleId  
)
```

VB

```
Public Function FindByIdAsync (  
    roleId As ULong  
) As Task(Of T)
```

C++

```
public:  
virtual Task<T>^ FindByIdAsync(  
    unsigned long long roleId  
) sealed
```

F#

```
abstract FindByIdAsync :  
    roleId : uint64 -> Task<'T>  
override FindByIdAsync :  
    roleId : uint64 -> Task<'T>
```

Parameters

roleId

Type: [System.UInt64](#)

[Missing <param name="roleId"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByIdAsync(System.UInt64)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByIdAsync(System.UInt64)"]

Implements

[IRoleStore.FindByIdAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).FindByNameAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByNameAsync(System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindByNameAsync(  
    string roleName  
)
```

VB

```
Public Function FindByNameAsync (  
    roleName As String  
) As Task(Of T)
```

C++

```
public:  
virtual Task<T>^ FindByNameAsync(  
    String^ roleName  
) sealed
```

F#

```
abstract FindByNameAsync :  
    roleName : string -> Task<'T>  
override FindByNameAsync :  
    roleName : string -> Task<'T>
```

Parameters

roleName

Type: [System.String](#)

[Missing <param name="roleName"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByNameAsync(System.String)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.FindByNameAsync(System.String)"]

Implements

[IRoleStore.FindByNameAsync\(String\)](#)

VelocityDB Class Library

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

RoleStore(T).UpdateAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.UpdateAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task UpdateAsync(  
    T role  
)
```

VB

```
Public Function UpdateAsync (  
    role As T  
) As Task
```

C++

```
public:  
virtual Task^ UpdateAsync(  
    T role  
) sealed
```

F#

```
abstract UpdateAsync :  
    role : 'T -> Task  
override UpdateAsync :  
    role : 'T -> Task
```

Parameters

role

Type: *T*

[Missing <param name="role"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.UpdateAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.RoleStore`1.UpdateAsync(`0)"]

Implements

IRoleStore.UpdateAsync(UTP)

VelocityDB Class Library

See Also

[RoleStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.UserLoginInfoAdapter"]

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityDBExtensions2.AspNet.Identity.UserLoginInfoAdapter

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class UserLoginInfoAdapter : OptimizedPersistable
```

VB

```
Public Class UserLoginInfoAdapter
    Inherits OptimizedPersistable
```

C++


```
public ref class UserLoginInfoAdapter : public OptimizedPersistable
```

F#



```
type UserLoginInfoAdapter =
    class
        inherit OptimizedPersistable
    end
```

The **UserLoginInfoAdapter** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------|
|  | UserLoginInfoAdapter | Initializes a new instance of the UserLoginInfoAdapter class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | LoginInfo | |
|  | UserId | |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter Constructor

Initializes a new instance of the [UserLoginInfoAdapter](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public UserLoginInfoAdapter ()
```

VB

```
Public Sub New
```

C++

```
public:  
UserLoginInfoAdapter ()
```

F#

```
new : unit -> UserLoginInfoAdapter
```

See Also



[UserLoginInfoAdapter Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter.UserLoginInfoAdapter Properties

The [UserLoginInfoAdapter](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | LoginInfo | |
|  | UserId | |

See Also

[UserLoginInfoAdapter Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter.LoginInfo Property

[Missing <summary> documentation for
"P:VelocityDBExtensions2.AspNet.Identity.UserLoginInfoAdapter.LoginInfo"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public UserLoginInfo LoginInfo { get; set; }
```

VB

```
Public Property LoginInfo As UserLoginInfo  
    Get  
    Set
```

C++

```
public:  
property UserLoginInfo^ LoginInfo {  
    UserLoginInfo^ get ();  
    void set (UserLoginInfo^ value);  
}
```

F#

```
member LoginInfo : UserLoginInfo with get, set
```

Property Value

Type: **UserLoginInfo**

See Also

[UserLoginInfoAdapter Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter.UserId Property

[Missing <summary> documentation for

"P:VelocityDBExtensions2.AspNet.Identity.UserLoginInfoAdapter.UserId"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public ulong UserId { get; set; }
```

VB

```
Public Property UserId As ULong  
    Get  
    Set
```

C++

```
public:  
property unsigned long long UserId {  
    unsigned long long get ();  
    void set (unsigned long long value);  
}
```

F#

```
member UserId : uint64 with get, set
```

Property Value

Type: [UInt64](#)

See Also



[UserLoginInfoAdapter Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserLoginInfoAdapter.UserLoginInfoAdapter Methods

The [UserLoginInfoAdapter](#) type exposes the following members.

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[UserLoginInfoAdapter Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T) Class

[Missing <summary> documentation for "T:VelocityDBExtensions2.AspNet.Identity.UserStore`1"]

Inheritance Hierarchy

[System.Object](#)

VelocityDBExtensions2.AspNet.Identity.UserStore(T)

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public class UserStore<T> : IUserLoginStore<T, ulong>,
    IUserStore<T, ulong>, IDisposable, IUserClaimStore<T, ulong>,
    IUserRoleStore<T, ulong>, IUserSecurityStampStore<T, ulong>,
    IQueryableUserStore<T, ulong>,
    IUserPasswordStore<T, ulong>, IUserPhoneNumberStore<T, ulong>,
    IUserLockoutStore<T, ulong>,
    IUserTwoFactorStore<T, ulong>, IUserEmailStore<T, ulong>
where T : IdentityUser
```

VB

```
Public Class UserStore(Of T As IdentityUser)
    Implements IUserLoginStore(Of T, ULong), IUserStore(Of T, ULong),
    IDisposable, IUserClaimStore(Of T, ULong), IUserRoleStore(Of T, ULong),
    IUserSecurityStampStore(Of T, ULong), IQueryableUserStore(Of T, ULong),
    IUserPasswordStore(Of T, ULong), IUserPhoneNumberStore(Of T, ULong),
    IUserLockoutStore(Of T, ULong), IUserTwoFactorStore(Of T, ULong),
    IUserEmailStore(Of T, ULong)
```

C++

```
generic<typename T>
where T : IdentityUser
public ref class UserStore : IUserLoginStore<T, unsigned long long>,
    IUserStore<T, unsigned long long>, IDisposable,
    IUserClaimStore<T, unsigned long long>, IUserRoleStore<T, unsigned long
long>,
    IUserSecurityStampStore<T, unsigned long long>, IQueryableUserStore<T,
unsigned long long>,
    IUserPasswordStore<T, unsigned long long>, IUserPhoneNumberStore<T,
unsigned long long>,
    IUserLockoutStore<T, unsigned long long>, IUserTwoFactorStore<T,
unsigned long long>,
    IUserEmailStore<T, unsigned long long>
```

```

F#
type UserStore<'T when 'T : IdentityUser> =
    class
        interface IUserLoginStore<'T, uint64>
        interface IUserStore<'T, uint64>
        interface IDisposable
        interface IUserClaimStore<'T, uint64>
        interface IUserRoleStore<'T, uint64>
        interface IUserSecurityStampStore<'T, uint64>
        interface IQueryableUserStore<'T, uint64>
        interface IUserPasswordStore<'T, uint64>
        interface IUserPhoneNumberStore<'T, uint64>
        interface IUserLockoutStore<'T, uint64>
        interface IUserTwoFactorStore<'T, uint64>
        interface IUserEmailStore<'T, uint64>
    end

```


Type Parameters

T


[Missing <typeparam name="T"/> documentation for "T:VelocityDBExtensions2.AspNet.Identity.UserStore`1"]

The UserStore(T) type exposes the following members.










Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|------------------------------------------------------|
|  | UserStore(T) | Initializes a new instance of the UserStore(T) class |
































Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|-------------|
|  | Users | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------------------------------------------|
|  | AddClaimAsync | |
|  | AddLoginAsync | |
|  | AddToRoleAsync | |
|  | CreateAsync | |
|  | DeleteAsync | |
|  | Dispose | Releases all resources used by the UserStore(T) |
|  | FindAsync | |
|  | FindByEmailAsync | |
|  | FindByIdAsync | |

VelocityDB Class Library

| | |
|-----------------------------------------------------------------------------------------------------------------------------------|--|
|  FindByNameAsync | |
|  GetAccessFailedCountAsync | |
|  GetClaimsAsync | |
|  GetEmailAsync | |
|  GetEmailConfirmedAsync | |
|  GetLockoutEnabledAsync | |
|  GetLockoutEndDateAsync | |
|  GetLoginsAsync | |
|  GetPasswordHashAsync | |
|  GetPhoneNumberAsync | |
|  GetPhoneNumberConfirmedAsync | |
|  GetRolesAsync | |
|  GetSecurityStampAsync | |
|  GetTwoFactorEnabledAsync | |
|  HasPasswordAsync | |
|  IncrementAccessFailedCountAsync | |
|  IsInRoleAsync | |
|  RemoveClaimAsync | |
|  RemoveFromRoleAsync | |
|  RemoveLoginAsync | |
|  ResetAccessFailedCountAsync | |
|  SetEmailAsync | |
|  SetEmailConfirmedAsync | |
|  SetLockoutEnabledAsync | |
|  SetLockoutEndDateAsync | |
|  SetPasswordHashAsync | |
|  SetPhoneNumberAsync | |
|  SetPhoneNumberConfirmedAsync | |
|  SetSecurityStampAsync | |
|  SetTwoFactorEnabledAsync | |
|  UpdateAsync | |

See Also

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T) Constructor

Initializes a new instance of the [UserStore\(T\)](#) class

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public UserStore(  
    AspNetIdentity identity  
)
```

VB

```
Public Sub New (  
    identity As AspNetIdentity  
)
```

C++

```
public:  
UserStore(  
    AspNetIdentity^ identity  
)
```

F#

```
new :  
    identity : AspNetIdentity -> UserStore
```

Parameters

identity

Type: [VelocityDBExtensions2.AspNet.Identity.AspNetIdentity](#)

[Missing <param name="identity"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.#ctor(VelocityDBExtensions2.AspNet.Identity.AspNetIdentity)"]

See Also


[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).UserStore(T) Properties

The [UserStore\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Users | |

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).Users Property

[Missing <summary> documentation for "P:VelocityDBExtensions2.AspNet.Identity.UserStore`1.Users"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public IQueryable<T> Users { get; }
```

VB

```
Public ReadOnly Property Users As IQueryable(Of T)  
    Get
```

C++

```
public:  
virtual property IQueryable<T>^ Users {  
    IQueryable<T>^ get () sealed;  
}
```

F#

```
abstract Users : IQueryable<'T> with get  
override Users : IQueryable<'T> with get
```

Property Value

Type: [IQueryable\(T\)](#)

Implements

[IQueryableUserStore.Users\(\)](#)

See Also





























[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).UserStore(T) Methods

The [UserStore\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------|-------------|
|  | AddClaimAsync | |
|  | AddLoginAsync | |
|  | AddToRoleAsync | |
|  | CreateAsync | |
|  | DeleteAsync | |
|  | Dispose | |
|  | FindAsync | |
|  | FindByEmailAsync | |
|  | FindByIdAsync | |
|  | FindByNameAsync | |
|  | GetAccessFailedCountAsync | |
|  | GetClaimsAsync | |
|  | GetEmailAsync | |
|  | GetEmailConfirmedAsync | |
|  | GetLockoutEnabledAsync | |
|  | GetLockoutEndDateAsync | |
|  | GetLoginsAsync | |
|  | GetPasswordHashAsync | |
|  | GetPhoneNumberAsync | |
|  | GetPhoneNumberConfirmedAsync | |
|  | GetRolesAsync | |
|  | GetSecurityStampAsync | |
|  | GetTwoFactorEnabledAsync | |
|  | HasPasswordAsync | |
|  | IncrementAccessFailedCountAsync | |
|  | IsInRoleAsync | |
|  | RemoveClaimAsync | |
|  | RemoveFromRoleAsync | |
|  | RemoveLoginAsync | |
|  | ResetAccessFailedCountAsync | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|----------------------------------------------|--|
|  | SetEmailAsync | |
|  | SetEmailConfirmedAsync | |
|  | SetLockoutEnabledAsync | |
|  | SetLockoutEndDateAsync | |
|  | SetPasswordHashAsync | |
|  | SetPhoneNumberAsync | |
|  | SetPhoneNumberConfirmedAsync | |
|  | SetSecurityStampAsync | |
|  | SetTwoFactorEnabledAsync | |
|  | UpdateAsync | |

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).AddClaimAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddClaimAsync(`0,System.Security.Claims.Claim)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task AddClaimAsync(  
    T user,  
    Claim claim  
)
```

VB

```
Public Function AddClaimAsync (  
    user As T,  
    claim As Claim  
) As Task
```

C++

```
public:  
virtual Task^ AddClaimAsync(  
    T user,  
    Claim^ claim  
) sealed
```

F#

```
abstract AddClaimAsync :  
    user : 'T *  
    claim : Claim -> Task  
override AddClaimAsync :  
    user : 'T *  
    claim : Claim -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddClaimAsync(`0,System.Security.Claims.Claim)"]

claim

Type: [System.Security.Claims.Claim](#)

[Missing <param name="claim"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddClaimAsync(`0,System.Security.Claims.Claim)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddClaimAsync(`0,System.Security.Claims.Claim)"]

Implements

IUserClaimStore.AddClaimAsync(UTP, Claim)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).AddLoginAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task AddLoginAsync(  
    T user,  
    UserLoginInfo login  
)
```

VB

```
Public Function AddLoginAsync (  
    user As T,  
    login As UserLoginInfo  
) As Task
```

C++

```
public:  
virtual Task^ AddLoginAsync(  
    T user,  
    UserLoginInfo^ login  
) sealed
```

F#

```
abstract AddLoginAsync :  
    user : 'T *  
    login : UserLoginInfo -> Task  
override AddLoginAsync :  
    user : 'T *  
    login : UserLoginInfo -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

login

Type: **UserLoginInfo**

[Missing <param name="login"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Implements

IUserLoginStore.AddLoginAsync(UTP, UserLoginInfo)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).AddToRoleAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddToRoleAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task AddToRoleAsync(  
    T user,  
    string roleName  
)
```

VB

```
Public Function AddToRoleAsync (  
    user As T,  
    roleName As String  
) As Task
```

C++

```
public:  
virtual Task^ AddToRoleAsync(  
    T user,  
    String^ roleName  
) sealed
```

F#

```
abstract AddToRoleAsync :  
    user : 'T *  
    roleName : string -> Task  
override AddToRoleAsync :  
    user : 'T *  
    roleName : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddToRoleAsync(`0,System.String)"]

roleName

Type: [System.String](#)

[Missing <param name="roleName"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddToRoleAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.AddToRoleAsync(`0,System.String)"]

Implements

IUserRoleStore.AddToRoleAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).CreateAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.CreateAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task CreateAsync(  
    T user  
)
```

VB

```
Public Function CreateAsync (  
    user As T  
) As Task
```

C++

```
public:  
virtual Task^ CreateAsync(  
    T user  
) sealed
```

F#

```
abstract CreateAsync :  
    user : 'T -> Task  
override CreateAsync :  
    user : 'T -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.CreateAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.CreateAsync(`0)"]

Implements

IUserStore.CreateAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).DeleteAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.DeleteAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task DeleteAsync(  
    T user  
)
```

VB

```
Public Function DeleteAsync (  
    user As T  
) As Task
```

C++

```
public:  
virtual Task^ DeleteAsync(  
    T user  
) sealed
```

F#

```
abstract DeleteAsync :  
    user : 'T -> Task  
override DeleteAsync :  
    user : 'T -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.DeleteAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.DeleteAsync(`0)"]

Implements

IUserStore.DeleteAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).Dispose Method

Releases all resources used by the [UserStore\(T\)](#)

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).FindAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindAsync(Microsoft.AspNet.Identity.UserLoginInfo)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindAsync (
    UserLoginInfo login
)
```

VB

```
Public Function FindAsync (
    login As UserLoginInfo
) As Task(Of T)
```

C++

```
public:
virtual Task<T>^ FindAsync (
    UserLoginInfo^ login
) sealed
```

F#

```
abstract FindAsync :
    login : UserLoginInfo -> Task<'T>
override FindAsync :
    login : UserLoginInfo -> Task<'T>
```

Parameters

login

Type: **UserLoginInfo**

[Missing <param name="login"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindAsync(Microsoft.AspNet.Identity.UserLoginInfo)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindAsync(Microsoft.AspNet.Identity.UserLoginInfo)"]

VelocityDB Class Library

Implements

IUserLoginStore.FindAsync(UserLoginInfo)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).FindByEmailAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByEmailAsync(System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindByEmailAsync(  
    string email  
)
```

VB

```
Public Function FindByEmailAsync (  
    email As String  
) As Task(Of T)
```

C++

```
public:  
virtual Task<T>^ FindByEmailAsync(  
    String^ email  
) sealed
```

F#

```
abstract FindByEmailAsync :  
    email : string -> Task<'T>  
override FindByEmailAsync :  
    email : string -> Task<'T>
```

Parameters

email

Type: [System.String](#)

[Missing <param name="email"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByEmailAsync(System.String)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByEmailAsync(System.String)"]

Implements

IUserEmailStore.FindByEmailAsync(String)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).FindByIdAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByIdAsync(System.UInt64)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindByIdAsync(  
    ulong userId  
)
```

VB

```
Public Function FindByIdAsync (  
    userId As ULong  
) As Task(Of T)
```

C++

```
public:  
virtual Task<T>^ FindByIdAsync(  
    unsigned long long userId  
) sealed
```

F#

```
abstract FindByIdAsync :  
    userId : uint64 -> Task<'T>  
override FindByIdAsync :  
    userId : uint64 -> Task<'T>
```

Parameters

userId

Type: [System.UInt64](#)

[Missing <param name="userId"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByIdAsync(System.UInt64)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByIdAsync(System.UInt64)"]

Implements

IUserStore.FindByIdAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).FindByNameAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByNameAsync(System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<T> FindByNameAsync (
    string userName
)
```

VB

```
Public Function FindByNameAsync (
    userName As String
) As Task(Of T)
```

C++

```
public:
virtual Task<T>^ FindByNameAsync (
    String^ userName
) sealed
```

F#

```
abstract FindByNameAsync :
    userName : string -> Task<'T>
override FindByNameAsync :
    userName : string -> Task<'T>
```

Parameters

userName

Type: [System.String](#)

[Missing <param name="userName"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByNameAsync(System.String)"]

Return Value

Type: [Task\(T\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.FindByNameAsync(System.String)"]

Implements

IUserStore.FindByNameAsync(String)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetAccessFailedCountAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetAccessFailedCountAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<int> GetAccessFailedCountAsync (  
    T user  
)
```

VB

```
Public Function GetAccessFailedCountAsync (  
    user As T  
) As Task(Of Integer)
```

C++

```
public:  
virtual Task<int>^ GetAccessFailedCountAsync (  
    T user  
) sealed
```

F#

```
abstract GetAccessFailedCountAsync :  
    user : 'T -> Task<int>  
override GetAccessFailedCountAsync :  
    user : 'T -> Task<int>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetAccessFailedCountAsync(`0)"]

Return Value

Type: [Task\(Int32\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetAccessFailedCountAsync(`0)"]

Implements

[IUserLockoutStore.GetAccessFailedCountAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetClaimsAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetClaimsAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<IList<Claim>> GetClaimsAsync(  
    T user  
)
```

VB

```
Public Function GetClaimsAsync (  
    user As T  
) As Task(Of IList(Of Claim))
```

C++

```
public:  
virtual Task<IList<Claim^>^> GetClaimsAsync(  
    T user  
) sealed
```

F#

```
abstract GetClaimsAsync :  
    user : 'T -> Task<IList<Claim>>  
override GetClaimsAsync :  
    user : 'T -> Task<IList<Claim>>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetClaimsAsync(`0)"]

Return Value

Type: [Task\(IList\(Claim\)\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetClaimsAsync(`0)"]

Implements

IUserClaimStore.GetClaimsAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetEmailAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<string> GetEmailAsync(  
    T user  
)
```

VB

```
Public Function GetEmailAsync (  
    user As T  
) As Task(Of String)
```

C++

```
public:  
virtual Task<String^> GetEmailAsync(  
    T user  
) sealed
```

F#

```
abstract GetEmailAsync :  
    user : 'T -> Task<string>  
override GetEmailAsync :  
    user : 'T -> Task<string>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailAsync(`0)"]

Return Value

Type: [Task\(String\)](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailAsync(`0)"]

Implements

IUserEmailStore.GetEmailAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetEmailConfirmedAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailConfirmedAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> GetEmailConfirmedAsync (  
    T user  
)
```

VB

```
Public Function GetEmailConfirmedAsync (  
    user As T  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ GetEmailConfirmedAsync (  
    T user  
) sealed
```

F#

```
abstract GetEmailConfirmedAsync :  
    user : 'T -> Task<bool>  
override GetEmailConfirmedAsync :  
    user : 'T -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailConfirmedAsync(`0)"]

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetEmailConfirmedAsync(`0)"]

Implements

IUserEmailStore.GetEmailConfirmedAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetLockoutEnabledAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEnabledAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> GetLockoutEnabledAsync (  
    T user  
)
```

VB

```
Public Function GetLockoutEnabledAsync (  
    user As T  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ GetLockoutEnabledAsync (  
    T user  
) sealed
```

F#

```
abstract GetLockoutEnabledAsync :  
    user : 'T -> Task<bool>  
override GetLockoutEnabledAsync :  
    user : 'T -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEnabledAsync(`0)"]

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEnabledAsync(`0)"]

Implements

[IUserLockoutStore.GetLockoutEnabledAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetLockoutEndDateAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEndDateAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<DateTimeOffset> GetLockoutEndDateAsync (
    T user
)
```

VB

```
Public Function GetLockoutEndDateAsync (
    user As T
) As Task(Of DateTimeOffset)
```

C++

```
public:
virtual Task<DateTimeOffset>^ GetLockoutEndDateAsync (
    T user
) sealed
```

F#

```
abstract GetLockoutEndDateAsync :
    user : 'T -> Task<DateTimeOffset>
override GetLockoutEndDateAsync :
    user : 'T -> Task<DateTimeOffset>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEndDateAsync(`0)"]

Return Value

Type: [Task\(DateTimeOffset\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLockoutEndDateAsync(`0)"]

Implements

IUserLockoutStore.GetLockoutEndDateAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetLoginsAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLoginsAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<IList<UserLoginInfo>> GetLoginsAsync (  
    T user  
)
```

VB

```
Public Function GetLoginsAsync (  
    user As T  
) As Task(Of IList(Of UserLoginInfo))
```

C++

```
public:  
virtual Task<IList<UserLoginInfo^>> GetLoginsAsync (  
    T user  
) sealed
```

F#

```
abstract GetLoginsAsync :  
    user : 'T -> Task<IList<UserLoginInfo>>  
override GetLoginsAsync :  
    user : 'T -> Task<IList<UserLoginInfo>>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLoginsAsync(`0)"]

Return Value

Type: [Task\(IList\(UserLoginInfo\)\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetLoginsAsync(`0)"]

Implements

[IUserLoginStore.GetLoginsAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetPasswordHashAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPasswordHashAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<string> GetPasswordHashAsync (
    T user
)
```

VB

```
Public Function GetPasswordHashAsync (
    user As T
) As Task(Of String)
```

C++

```
public:
virtual Task<String^>^ GetPasswordHashAsync (
    T user
) sealed
```

F#

```
abstract GetPasswordHashAsync :
    user : 'T -> Task<string>
override GetPasswordHashAsync :
    user : 'T -> Task<string>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPasswordHashAsync(`0)"]

Return Value

Type: [Task\(String\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPasswordHashAsync(`0)"]

Implements

IUserPasswordStore.GetPasswordHashAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetPhoneNumberAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<string> GetPhoneNumberAsync (  
    T user  
)
```

VB

```
Public Function GetPhoneNumberAsync (  
    user As T  
) As Task(Of String)
```

C++

```
public:  
virtual Task<String^>^ GetPhoneNumberAsync (  
    T user  
) sealed
```

F#

```
abstract GetPhoneNumberAsync :  
    user : 'T -> Task<string>  
override GetPhoneNumberAsync :  
    user : 'T -> Task<string>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberAsync(`0)"]

Return Value

Type: [Task\(String\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberAsync(`0)"]

Implements

[IUserPhoneNumberStore.GetPhoneNumberAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetPhoneNumberConfirmedAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberConfirmedAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> GetPhoneNumberConfirmedAsync (  
    T user  
)
```

VB

```
Public Function GetPhoneNumberConfirmedAsync (  
    user As T  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ GetPhoneNumberConfirmedAsync (  
    T user  
) sealed
```

F#

```
abstract GetPhoneNumberConfirmedAsync :  
    user : 'T -> Task<bool>  
override GetPhoneNumberConfirmedAsync :  
    user : 'T -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberConfirmedAsync(`0)"]

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetPhoneNumberConfirmedAsync(`0)"]

Implements

IUserPhoneNumberStore.GetPhoneNumberConfirmedAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetRolesAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetRolesAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<IList<string>> GetRolesAsync(  
    T user  
)
```

VB

```
Public Function GetRolesAsync (  
    user As T  
) As Task(Of IList(Of String))
```

C++

```
public:  
virtual Task<IList<String^>> GetRolesAsync(  
    T user  
) sealed
```

F#

```
abstract GetRolesAsync :  
    user : 'T -> Task<IList<string>>  
override GetRolesAsync :  
    user : 'T -> Task<IList<string>>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetRolesAsync(`0)"]

Return Value

Type: [Task\(IList\(String\)\)](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetRolesAsync(`0)"]

Implements

IUserRoleStore.GetRolesAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetSecurityStampAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetSecurityStampAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<string> GetSecurityStampAsync (  
    T user  
)
```

VB

```
Public Function GetSecurityStampAsync (  
    user As T  
) As Task(Of String)
```

C++

```
public:  
virtual Task<String^>^ GetSecurityStampAsync (  
    T user  
) sealed
```

F#

```
abstract GetSecurityStampAsync :  
    user : 'T -> Task<string>  
override GetSecurityStampAsync :  
    user : 'T -> Task<string>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetSecurityStampAsync(`0)"]

Return Value

Type: [Task\(String\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetSecurityStampAsync(`0)"]

Implements

IUserSecurityStampStore.GetSecurityStampAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).GetTwoFactorEnabledAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetTwoFactorEnabledAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> GetTwoFactorEnabledAsync (  
    T user  
)
```

VB

```
Public Function GetTwoFactorEnabledAsync (  
    user As T  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ GetTwoFactorEnabledAsync (  
    T user  
) sealed
```

F#

```
abstract GetTwoFactorEnabledAsync :  
    user : 'T -> Task<bool>  
override GetTwoFactorEnabledAsync :  
    user : 'T -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetTwoFactorEnabledAsync(`0)"]

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.GetTwoFactorEnabledAsync(`0)"]

Implements

[IUserTwoFactorStore.GetTwoFactorEnabledAsync\(UTP\)](#)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).HasPasswordAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.HasPasswordAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> HasPasswordAsync (  
    T user  
)
```

VB

```
Public Function HasPasswordAsync (  
    user As T  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ HasPasswordAsync (  
    T user  
) sealed
```

F#

```
abstract HasPasswordAsync :  
    user : 'T -> Task<bool>  
override HasPasswordAsync :  
    user : 'T -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.HasPasswordAsync(`0)"]

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.HasPasswordAsync(`0)"]

Implements

IUserPasswordStore.HasPasswordAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).IncrementAccessFailedCountAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IncrementAccessFailedCountAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<int> IncrementAccessFailedCountAsync (  
    T user  
)
```

VB

```
Public Function IncrementAccessFailedCountAsync (  
    user As T  
) As Task(Of Integer)
```

C++

```
public:  
virtual Task<int>^ IncrementAccessFailedCountAsync (  
    T user  
) sealed
```

F#

```
abstract IncrementAccessFailedCountAsync :  
    user : 'T -> Task<int>  
override IncrementAccessFailedCountAsync :  
    user : 'T -> Task<int>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IncrementAccessFailedCountAsync(`0)"]

Return Value

Type: [Task\(Int32\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IncrementAccessFailedCountAsync(`0)"]

Implements

IUserLockoutStore.IncrementAccessFailedCountAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).IsInRoleAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IsInRoleAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task<bool> IsInRoleAsync(  
    T user,  
    string roleName  
)
```

VB

```
Public Function IsInRoleAsync (  
    user As T,  
    roleName As String  
) As Task(Of Boolean)
```

C++

```
public:  
virtual Task<bool>^ IsInRoleAsync(  
    T user,  
    String^ roleName  
) sealed
```

F#

```
abstract IsInRoleAsync :  
    user : 'T *  
    roleName : string -> Task<bool>  
override IsInRoleAsync :  
    user : 'T *  
    roleName : string -> Task<bool>
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IsInRoleAsync(`0,System.String)"]

roleName

Type: [System.String](#)

[Missing <param name="roleName"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IsInRoleAsync(`0,System.String)"]

VelocityDB Class Library

Return Value

Type: [Task\(Boolean\)](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.IsInRoleAsync(`0,System.String)"]

Implements

IUserRoleStore.IsInRoleAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).RemoveClaimAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveClaimAsync(`0,System.Security.Claims.Claim)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task RemoveClaimAsync (  
    T user,  
    Claim claim  
)
```

VB

```
Public Function RemoveClaimAsync (  
    user As T,  
    claim As Claim  
) As Task
```

C++

```
public:  
virtual Task^ RemoveClaimAsync (  
    T user,  
    Claim^ claim  
) sealed
```

F#

```
abstract RemoveClaimAsync :  
    user : 'T *  
    claim : Claim -> Task  
override RemoveClaimAsync :  
    user : 'T *  
    claim : Claim -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveClaimAsync(`0,System.Security.Claims.Claim)"]

claim

Type: [System.Security.Claims.Claim](#)

[Missing <param name="claim"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveClaimAsync(`0,System.Security.Claims.Claim)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveClaimAsync(`0,System.Security.Claims.Claim)"]

Implements

IUserClaimStore.RemoveClaimAsync(UTP, Claim)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).RemoveFromRoleAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveFromRoleAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task RemoveFromRoleAsync(  
    T user,  
    string roleName  
)
```

VB

```
Public Function RemoveFromRoleAsync (  
    user As T,  
    roleName As String  
) As Task
```

C++

```
public:  
virtual Task^ RemoveFromRoleAsync(  
    T user,  
    String^ roleName  
) sealed
```

F#

```
abstract RemoveFromRoleAsync :  
    user : 'T *  
    roleName : string -> Task  
override RemoveFromRoleAsync :  
    user : 'T *  
    roleName : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveFromRoleAsync(`0,System.String)"]

roleName

Type: [System.String](#)

[Missing <param name="roleName"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveFromRoleAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveFromRoleAsync(`0,System.String)"]

Implements

IUserRoleStore.RemoveFromRoleAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).RemoveLoginAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task RemoveLoginAsync (  
    T user,  
    UserLoginInfo login  
)
```

VB

```
Public Function RemoveLoginAsync (  
    user As T,  
    login As UserLoginInfo  
) As Task
```

C++

```
public:  
virtual Task^ RemoveLoginAsync (  
    T user,  
    UserLoginInfo^ login  
) sealed
```

F#

```
abstract RemoveLoginAsync :  
    user : 'T *  
    login : UserLoginInfo -> Task  
override RemoveLoginAsync :  
    user : 'T *  
    login : UserLoginInfo -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

login

Type: **UserLoginInfo**

[Missing <param name="login"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.RemoveLoginAsync(`0,Microsoft.AspNet.Identity.UserLoginInfo)"]

Implements

IUserLoginStore.RemoveLoginAsync(UTP, UserLoginInfo)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).ResetAccessFailedCountAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.ResetAccessFailedCountAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task ResetAccessFailedCountAsync (  
    T user  
)
```

VB

```
Public Function ResetAccessFailedCountAsync (  
    user As T  
) As Task
```

C++

```
public:  
virtual Task^ ResetAccessFailedCountAsync (  
    T user  
) sealed
```

F#

```
abstract ResetAccessFailedCountAsync :  
    user : 'T -> Task  
override ResetAccessFailedCountAsync :  
    user : 'T -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.ResetAccessFailedCountAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.ResetAccessFailedCountAsync(`0)"]

Implements

IUserLockoutStore.ResetAccessFailedCountAsync(UTP)

VelocityDB Class Library

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetEmailAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetEmailAsync(  
    T user,  
    string email  
)
```

VB

```
Public Function SetEmailAsync (  
    user As T,  
    email As String  
) As Task
```

C++

```
public:  
virtual Task^ SetEmailAsync(  
    T user,  
    String^ email  
) sealed
```

F#

```
abstract SetEmailAsync :  
    user : 'T *  
    email : string -> Task  
override SetEmailAsync :  
    user : 'T *  
    email : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailAsync(`0,System.String)"]

email

Type: [System.String](#)

[Missing <param name="email"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailAsync(`0,System.String)"]

Implements

IUserEmailStore.SetEmailAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetEmailConfirmedAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailConfirmedAsync(`0,System.Boolean)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetEmailConfirmedAsync (  
    T user,  
    bool confirmed  
)
```

VB

```
Public Function SetEmailConfirmedAsync (  
    user As T,  
    confirmed As Boolean  
) As Task
```

C++

```
public:  
virtual Task^ SetEmailConfirmedAsync (  
    T user,  
    bool confirmed  
) sealed
```

F#

```
abstract SetEmailConfirmedAsync :  
    user : 'T *  
    confirmed : bool -> Task  
override SetEmailConfirmedAsync :  
    user : 'T *  
    confirmed : bool -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailConfirmedAsync(`0,System.Boolean)"]

confirmed

Type: [System.Boolean](#)

[Missing <param name="confirmed"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailConfirmedAsync(`0,System.Boolean)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetEmailConfirmedAsync(`0,System.Boolean)"]

Implements

IUserEmailStore.SetEmailConfirmedAsync(UTP, Boolean)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetLockoutEnabledAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEnabledAsync(`0,System.Boolean)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetLockoutEnabledAsync (  
    T user,  
    bool enabled  
)
```

VB

```
Public Function SetLockoutEnabledAsync (  
    user As T,  
    enabled As Boolean  
) As Task
```

C++

```
public:  
virtual Task^ SetLockoutEnabledAsync (  
    T user,  
    bool enabled  
) sealed
```

F#

```
abstract SetLockoutEnabledAsync :  
    user : 'T *  
    enabled : bool -> Task  
override SetLockoutEnabledAsync :  
    user : 'T *  
    enabled : bool -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEnabledAsync(`0,System.Boolean)"]

enabled

Type: [System.Boolean](#)

**[Missing <param name="enabled"/> documentation for
"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEnabledAsync(`0,System.Boolean)
"]**

Return Value

Type: [Task](#)

**[Missing <returns> documentation for
"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEnabledAsync(`0,System.Boolean)
"]**

Implements

IUserLockoutStore.SetLockoutEnabledAsync(UTP, Boolean)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetLockoutEndDateAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEndDateAsync(`0,System.DateTimeOffset)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetLockoutEndDateAsync (  
    T user,  
    DateTimeOffset lockoutEnd  
)
```

VB

```
Public Function SetLockoutEndDateAsync (  
    user As T,  
    lockoutEnd As DateTimeOffset  
) As Task
```

C++

```
public:  
virtual Task^ SetLockoutEndDateAsync (  
    T user,  
    DateTimeOffset lockoutEnd  
) sealed
```

F#

```
abstract SetLockoutEndDateAsync :  
    user : 'T *  
    lockoutEnd : DateTimeOffset -> Task  
override SetLockoutEndDateAsync :  
    user : 'T *  
    lockoutEnd : DateTimeOffset -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEndDateAsync(`0,System.DateTimeOffset)"]

lockoutEnd

Type: [System.DateTimeOffset](#)

[Missing <param name="lockoutEnd"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEndDateAsync(`0,System.DateTimeOffset)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetLockoutEndDateAsync(`0,System.DateTimeOffset)"]

Implements

IUserLockoutStore.SetLockoutEndDateAsync(UTP, DateTimeOffset)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetPasswordHashAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPasswordHashAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetPasswordHashAsync (  
    T user,  
    string passwordHash  
)
```

VB

```
Public Function SetPasswordHashAsync (  
    user As T,  
    passwordHash As String  
) As Task
```

C++

```
public:  
virtual Task^ SetPasswordHashAsync (  
    T user,  
    String^ passwordHash  
) sealed
```

F#

```
abstract SetPasswordHashAsync :  
    user : 'T *  
    passwordHash : string -> Task  
override SetPasswordHashAsync :  
    user : 'T *  
    passwordHash : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPasswordHashAsync(`0,System.String)"]

passwordHash

Type: [System.String](#)

[Missing <param name="passwordHash"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPasswordHashAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPasswordHashAsync(`0,System.String)"]

Implements

IUserPasswordStore.SetPasswordHashAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetPhoneNumberAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetPhoneNumberAsync (  
    T user,  
    string phoneNumber  
)
```

VB

```
Public Function SetPhoneNumberAsync (  
    user As T,  
    phoneNumber As String  
) As Task
```

C++

```
public:  
virtual Task^ SetPhoneNumberAsync (  
    T user,  
    String^ phoneNumber  
) sealed
```

F#

```
abstract SetPhoneNumberAsync :  
    user : 'T *  
    phoneNumber : string -> Task  
override SetPhoneNumberAsync :  
    user : 'T *  
    phoneNumber : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberAsync(`0,System.String)"]

phoneNumber

Type: [System.String](#)

[Missing <param name="phoneNumber"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberAsync(`0,System.String)"]

Implements

IUserPhoneNumberStore.SetPhoneNumberAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetPhoneNumberConfirmedAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberConfirmedAsync(`0,System.Boolean)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetPhoneNumberConfirmedAsync (  
    T user,  
    bool confirmed  
)
```

VB

```
Public Function SetPhoneNumberConfirmedAsync (  
    user As T,  
    confirmed As Boolean  
) As Task
```

C++

```
public:  
virtual Task^ SetPhoneNumberConfirmedAsync (  
    T user,  
    bool confirmed  
) sealed
```

F#

```
abstract SetPhoneNumberConfirmedAsync :  
    user : 'T *  
    confirmed : bool -> Task  
override SetPhoneNumberConfirmedAsync :  
    user : 'T *  
    confirmed : bool -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberConfirmedAsync(`0,System.Boolean)"]

confirmed

Type: [System.Boolean](#)

[Missing <param name="confirmed"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberConfirmedAsync(`0,System.Boolean)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetPhoneNumberConfirmedAsync(`0,System.Boolean)"]

Implements

IUserPhoneNumberStore.SetPhoneNumberConfirmedAsync(UTP, Boolean)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetSecurityStampAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetSecurityStampAsync(`0,System.String)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetSecurityStampAsync (  
    T user,  
    string stamp  
)
```

VB

```
Public Function SetSecurityStampAsync (  
    user As T,  
    stamp As String  
) As Task
```

C++

```
public:  
virtual Task^ SetSecurityStampAsync (  
    T user,  
    String^ stamp  
) sealed
```

F#

```
abstract SetSecurityStampAsync :  
    user : 'T *  
    stamp : string -> Task  
override SetSecurityStampAsync :  
    user : 'T *  
    stamp : string -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetSecurityStampAsync(`0,System.String)"]

stamp

Type: [System.String](#)

[Missing <param name="stamp"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetSecurityStampAsync(`0,System.String)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetSecurityStampAsync(`0,System.String)"]

Implements

IUserSecurityStampStore.SetSecurityStampAsync(UTP, String)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).SetTwoFactorEnabledAsync Method

[Missing <summary> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetTwoFactorEnabledAsync(`0,System.Boolean)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task SetTwoFactorEnabledAsync (  
    T user,  
    bool enabled  
)
```

VB

```
Public Function SetTwoFactorEnabledAsync (  
    user As T,  
    enabled As Boolean  
) As Task
```

C++

```
public:  
virtual Task^ SetTwoFactorEnabledAsync (  
    T user,  
    bool enabled  
) sealed
```

F#

```
abstract SetTwoFactorEnabledAsync :  
    user : 'T *  
    enabled : bool -> Task  
override SetTwoFactorEnabledAsync :  
    user : 'T *  
    enabled : bool -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for

"M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetTwoFactorEnabledAsync(`0,System.Boolean)"]

enabled

Type: [System.Boolean](#)

[Missing <param name="enabled"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetTwoFactorEnabledAsync(`0,System.Boolean)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.SetTwoFactorEnabledAsync(`0,System.Boolean)"]

Implements

IUserTwoFactorStore.SetTwoFactorEnabledAsync(UTP, Boolean)

See Also

[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)

UserStore(T).UpdateAsync Method

[Missing <summary> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.UpdateAsync(`0)"]

Namespace: [VelocityDBExtensions2.AspNet.Identity](#)

Assembly: VelocityDBExtensions2 (in VelocityDBExtensions2.dll) Version: 11.1.0.0 (11.1.0)

Syntax

C#

```
public Task UpdateAsync(  
    T user  
)
```

VB

```
Public Function UpdateAsync (  
    user As T  
) As Task
```

C++

```
public:  
virtual Task^ UpdateAsync(  
    T user  
) sealed
```

F#

```
abstract UpdateAsync :  
    user : 'T -> Task  
override UpdateAsync :  
    user : 'T -> Task
```

Parameters

user

Type: *T*

[Missing <param name="user"/> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.UpdateAsync(`0)"]

Return Value

Type: [Task](#)

[Missing <returns> documentation for "M:VelocityDBExtensions2.AspNet.Identity.UserStore`1.UpdateAsync(`0)"]

Implements

IUserStore.UpdateAsync(UTP)

VelocityDB Class Library

See Also











[UserStore\(T\)Class](#)

[VelocityDBExtensions2.AspNet.Identity Namespace](#)



VelocityGraph Namespace

The **VelocityGraph** namespace contains classes for the VelocityGraph Graph Database.

Classes

| Class | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Edge | An Edge links two vertices. Along with its key/value properties, an edge has both a directionality and a label. The directionality determines which vertex is the tail vertex (out vertex) and which vertex is the head vertex (in vertex). The edge label determines the type of relationship that exists between the two vertices. Diagrammatically, outVertex ---label---> inVertex. |
|  EdgeType | All edges have a type that is identified as a EdgeType. Each EdgeType have a name, can be persisted and tracks all edges of its type. The edge type also knows about properties used by its type. |
|  Element | Base class for Edge and Vertex |
|  Graph | Graph is the root object of a graph. Most graph api is on this class but useful api also exist on VertexType and EdgeType . |
|  GraphJsonSettings | |
|  PropertyType | Keeper of Edge and Vertex properties |
|  PropertyTypeNoDuplicateValues(T) | Used with string property type values. Avoids storing duplicate string values. |
|  PropertyTypeT(T) | Keeper of Edge and Vertex properties |
|  Vertex | A vertex maintains pointers to both a set of incoming and outgoing edges. The outgoing edges are those edges for which the vertex is the tail. The incoming edges are those edges for which the vertex is the head. Diagrammatically, ---inEdges---> vertex ---outEdges---> |
|  VertexType | All vertices have a type that is identified as a VertexType. Each VertexType have a name, can be persisted and tracks all vertices of its type. The vertex type also knows about properties used by its type. |

Enumerations

| Enumeration | Description |
|------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DataType | Choose IOptimizedPersistable instead of Object for property types where you only will store objects of type IOptimizedPersistable as property value. |
|  PropertyKind | A property can be index or not or index can require each entry to be unique |

Data Type Enumeration

Choose [IOptimizedPersistable](#) instead of [Object](#) for property types where you only will store objects of type IOptimizedPersistable as property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum DataType
```

VB

```
Public Enumeration DataType
```

C++

```
public enum class DataType
```

F#

```
type DataType
```

Members

| Member name | Value | Description |
|------------------------------|-------|-----------------------------------------------------|
| Boolean | 0 | Correspond to Boolean |
| Integer | 1 | Correspond to Int32 |
| Long | 2 | Correspond to Int64 |
| Double | 3 | Correspond to Double |
| Single | 4 | Correspond to Single |
| DateTime | 5 | Correspond to DateTime |
| String | 6 | Correspond to String |
| Object | 7 | Correspond to Object |
| IOptimizedPersistable | 8 | Correspond to IOptimizedPersistable |

See Also

[VelocityGraph Namespace](#)

Edge Class

An Edge links two vertices. Along with its key/value properties, an edge has both a directionality and a label. The directionality determines which vertex is the tail vertex (out vertex) and which vertex is the head vertex (in vertex). The edge label determines the type of relationship that exists between the two vertices. Diagrammatically, outVertex ---label--> inVertex.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Element](#)

VelocityGraph.Edge

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Edge : Element, IEdge, IElement,
    IDictionary<string, Object>, ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection, IEqualityComparer<Edge>
```

VB

```
Public Class Edge
    Inherits Element
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
    KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection, IEqualityComparer(Of Edge)
```

C++

```
public ref class Edge : public Element,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection, IEqualityComparer<Edge^>
```

F#

```
type Edge =
    class
        inherit Element
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
```






```

interface IDictionary
interface ICollection
interface IEqualityComparer<Edge>
end












```



The **Edge** type exposes the following members.

Properties











| Name | Description |
|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  EdgeId | Gets the edge id |
|  EdgeType | Gets the edge type |
|  Head | Gets the head vertex |
|  Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. (Overrides DictionaryElement.Id.) |
|  Tail | Gets the tail vertex |

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Equals(Object) | Determines whether the specified Edge is equal to the current one. (Overrides Object.Equals(Object).) |
|  Equals(Edge, Edge) | Determines whether the specified objects are equal. |
|  GetEdgePeer | Gets the other end for the given edge. |
|  GetHashCode() | Use id as hash code (Overrides Element.GetHashCode().) |
|  GetHashCode(Edge) | Returns a hash code for the specified object. |
|  GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String).) |
|  GetProperty(PropertyType) | Gets the Value for the given Property id |
|  GetPropertyKeys | Return all the keys associated with the element. (Overrides DictionaryElement.GetPropertyKeys().) |
|  Remove | Remove the edge from the graph. (Overrides DictionaryElement.Remove().) |
|  RemoveProperty | Un-assigns a key/value property from the edge. The object value of the removed property is returned. (Overrides DictionaryElement.RemoveProperty(String).) |
|  SetProperty(String, Object) | Assign a key/value property to the edge. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object).) |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
|  SetProperty(PropertyType, IComparable) | Sets the value for a property |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  EdgeString | (Defined by StringFactory .) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |






See Also

[VelocityGraph Namespace](#)

Edge.Edge Properties

The [Edge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | EdgeId | Gets the edge id |
|  | EdgeType | Gets the edge type |
|  | Head | Gets the head vertex |
|  | Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. (Overrides DictionaryElement.Id.) |
|  | Tail | Gets the tail vertex |

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.EdgeId Property

Gets the edge id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int EdgeId { get; }
```

VB

```
Public ReadOnly Property EdgeId As Integer  
    Get
```

C++

```
public:  
property int EdgeId {  
    int get ();  
}
```

F#

```
member EdgeId : int with get
```

Property Value

Type: [Int32](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.EdgeType Property

Gets the edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType EdgeType { get; }
```

VB

```
Public ReadOnly Property EdgeType As EdgeType  
    Get
```

C++

```
public:  
property EdgeType^ EdgeType {  
    EdgeType^ get ();  
}
```

F#

```
member EdgeType : EdgeType with get
```

Property Value

Type: [EdgeType](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Head Property

Gets the head vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex Head { get; }
```

VB

```
Public ReadOnly Property Head As Vertex  
    Get
```

C++

```
public:  
property Vertex^ Head {  
    Vertex^ get ();  
}
```

F#

```
member Head : Vertex with get
```

Property Value

Type: [Vertex](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Id Property

An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Return Value

Type: [Object](#)

the identifier of the element

Implements

[IElement.Id](#)

[IElement.Id](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Tail Property

Gets the tail vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex Tail { get; }
```

VB

```
Public ReadOnly Property Tail As Vertex  
    Get
```

C++

```
public:  
property Vertex^ Tail {  
    Vertex^ get ();  
}
```

F#

```
member Tail : Vertex with get
```

Property Value

Type: [Vertex](#)

See Also














[Edge Class](#)

[VelocityGraph Namespace](#)



Edge.Edge Methods









The [Edge](#) type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Equals(Object) | Determines whether the specified Edge is equal to the current one. (Overrides Object.Equals(Object) .) |
|  Equals(Edge, Edge) | Determines whether the specified objects are equal. |
|  GetEdgePeer | Gets the other end for the given edge. |
|  GetHashCode() | Use id as hash code (Overrides Element.GetHashCode() .) |
|  GetHashCode(Edge) | Returns a hash code for the specified object. |
|  GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String) .) |
|  GetProperty(PropertyType) | Gets the Value for the given Property id |
|  GetPropertyKeys | Return all the keys associated with the element. (Overrides DictionaryElement.GetPropertyKeys() .) |
|  Remove | Remove the edge from the graph. (Overrides DictionaryElement.Remove() .) |
|  RemoveProperty | Un-assigns a key/value property from the edge. The object value of the removed property is returned. (Overrides DictionaryElement.RemoveProperty(String) .) |
|  SetProperty(String, Object) | Assign a key/value property to the edge. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  SetProperty(PropertyType, IComparable) | Sets the value for a property |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |

Extension Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  EdgeString | (Defined by StringFactory .) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |



See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Equals Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
|  | Equals(Object) | Determines whether the specified Edge is equal to the current one. (Overrides Object.Equals(Object) .) |
|  | Equals(Edge, Edge) | Determines whether the specified objects are equal. |

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Equals Method (Object)

Determines whether the specified [Edge](#) is equal to the current one.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object other  
)
```

VB

```
Public Overrides Function Equals (  
    other As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ other  
) override
```

F#

```
abstract Equals :  
    other : Object -> bool  
override Equals :  
    other : Object -> bool
```

Parameters

other

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` if the specified object is equal to the current object; otherwise, `false`.

See Also

[Edge Class](#)

[Equals Overload](#)

[VelocityGraph Namespace](#)

Edge.Equals Method (Edge, Edge)

Determines whether the specified objects are equal.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    Edge x,  
    Edge y  
)
```

VB

```
Public Function Equals (  
    x As Edge,  
    y As Edge  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Edge^ x,  
    Edge^ y  
) sealed
```

F#

```
abstract Equals :  
    x : Edge *  
    y : Edge -> bool  
override Equals :  
    x : Edge *  
    y : Edge -> bool
```

Parameters

x

Type: [VelocityGraph.Edge](#)

The first object of type *T* to compare.

y

Type: [VelocityGraph.Edge](#)

The second object of type *T* to compare.

Return Value

Type: [Boolean](#)

`true` (True in Visual Basic) if the specified objects are equal; otherwise, `false` (False in Visual Basic).

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)

See Also

[Edge Class](#)

[Equals Overload](#)

[VelocityGraph Namespace](#)

Edge.GetEdgePeer Method

Gets the other end for the given edge.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex GetEdgePeer (
    Vertex vertex
)
```

VB

```
Public Function GetEdgePeer (
    vertex As Vertex
) As Vertex
```

C++

```
public:
Vertex^ GetEdgePeer (
    Vertex^ vertex
)
```

F#

```
member GetEdgePeer :
    vertex : Vertex -> Vertex
```

Parameters

vertex

Type: [VelocityGraph.Vertex](#)

A vertex, it must be one of the ends of the edge.

Return Value

Type: [Vertex](#)

The other end of the edge.



See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.GetHashCode Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------------------------------------------------------------------|
|  | GetHashCode() | Use id as hash code (Overrides Element.GetHashCode() .) |
|  | GetHashCode(Edge) | Returns a hash code for the specified object. |

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.GetHashCode Method

Use id as hash code

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

The hash code given by id

See Also

[Edge Class](#)

[GetHashCode Overload](#)

[VelocityGraph Namespace](#)

Edge.GetHashCode Method (Edge)

Returns a hash code for the specified object.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode(  
    Edge obj  
)
```

VB

```
Public Function GetHashCode (  
    obj As Edge  
) As Integer
```

C++

```
public:  
virtual int GetHashCode (  
    Edge^ obj  
) sealed
```

F#

```
abstract GetHashCode :  
    obj : Edge -> int  
override GetHashCode :  
    obj : Edge -> int
```

Parameters

obj

Type: [VelocityGraph.Edge](#)

The [Object](#) for which a hash code is to be returned.

Return Value

Type: [Int32](#)

A hash code for the specified object.

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| ArgumentNullException | The type of <i>obj</i> is a reference type and <i>obj</i> is a null reference (Nothing in Visual Basic). |

See Also



[Edge Class](#)

[GetHashCode Overload](#)

[VelocityGraph Namespace](#)

Edge.GetProperty Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetProperty(PropertyType) | Gets the Value for the given Property id |

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.GetProperty Method (String)

Return the object value associated with the provided string key. If no value exists for that key, return null.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the key/value property

Return Value

Type: [Object](#)

the object value related to the string key

Implements

[IElement.GetProperty\(String\)](#)

[IElement.GetProperty\(String\)](#)

See Also

[Edge Class](#)

VelocityDB Class Library

[GetProperty Overload](#)

[VelocityGraph Namespace](#)

Edge.GetProperty Method (PropertyType)

Gets the Value for the given Property id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IComparable GetProperty(  
    PropertyType property  
)
```

VB

```
Public Function GetProperty (  
    property As PropertyType  
) As IComparable
```

C++

```
public:  
IComparable^ GetProperty(  
    PropertyType^ property  
)
```

F#

```
member GetProperty :  
    property : PropertyType -> IComparable
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

Property type identifier.

Return Value

Type: [IComparable](#)

Property value as [IComparable](#)

See Also

[Edge Class](#)

[GetProperty Overload](#)

[VelocityGraph Namespace](#)

Edge.GetPropertyKeys Method

Return all the keys associated with the element.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

the set of all string keys associated with the element

Implements

[IElement.GetPropertyKeys\(\)](#)

[IElement.GetPropertyKeys\(\)](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.Remove Method

Remove the edge from the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

[IElement.Remove\(\)](#)

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.RemoveProperty Method

Un-assigns a key/value property from the edge. The object value of the removed property is returned.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the property to remove from the edge

Return Value

Type: [Object](#)

the object value associated with that key prior to removal

Implements

[IElement.RemoveProperty\(String\)](#)

[IElement.RemoveProperty\(String\)](#)



See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.SetProperty Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | SetProperty(String, Object) | Assign a key/value property to the edge. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | SetProperty(PropertyType, IComparable) | Sets the value for a property |

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

Edge.SetProperty Method (String, Object)

Assign a key/value property to the edge. If a value already exists for this key, then the previous key/value is overwritten.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

the string key of the property

value

Type: [System.Object](#)

the object value o the property

VelocityDB Class Library

Implements

[IElement.SetProperty\(String, Object\)](#)

[IElement.SetProperty\(String, Object\)](#)

See Also

[Edge Class](#)

[SetProperty Overload](#)

[VelocityGraph Namespace](#)

Edge.SetProperty Method (PropertyType, IComparable)

Sets the value for a property

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetProperty(  
    PropertyType property,  
    IComparable v  
)
```

VB

```
Public Sub SetProperty (  
    property As PropertyType,  
    v As IComparable  
)
```

C++

```
public:  
void SetProperty(  
    PropertyType^ property,  
    IComparable^ v  
)
```

F#

```
member SetProperty :  
    property : PropertyType *  
    v : IComparable -> unit
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

The property to set

v

Type: [System.IComparable](#)

the value

See Also

[Edge Class](#)

[SetProperty Overload](#)

[VelocityGraph Namespace](#)

Edge.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[Edge Class](#)

[VelocityGraph Namespace](#)

EdgeType Class

All edges have a type that is identified as a EdgeType. Each EdgeType have a name, can be persisted and tracks all edges of its type. The edge type also knows about properties used by its type.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityGraph.EdgeType

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class EdgeType : OptimizedPersistable, IComparable<EdgeType>,
    IEqualityComparer<EdgeType>
```

VB

```
<SerializableAttribute>
Public Class EdgeType
    Inherits OptimizedPersistable
    Implements IComparable(Of EdgeType), IEqualityComparer(Of EdgeType)
```

C++


```
[SerializableAttribute]
public ref class EdgeType : public OptimizedPersistable,
    IComparable<EdgeType^>, IEqualityComparer<EdgeType^>
```

F#









```
[<SerializableAttribute>]
type EdgeType =
    class
        inherit OptimizedPersistable
        interface IComparable<EdgeType>
        interface IEqualityComparer<EdgeType>
    end
```

The **EdgeType** type exposes the following members.


















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|--------------------------|
|  | EdgeType | Creates a new edge type. |





Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------|
|  | Directed | Is Directed if edge type is not bidirectional |
|  | HeadType | Gets the Head VertexType of the edge (might not be set) |
|  | MyGraph | Graph for which this EdgeType belongs to |
|  | SubTypes | Sub types of this EdgeType |
|  | TailType | Gets the Tail VertexType of the edge (might not be set) |
|  | TypeId | Gets the id of the edge type |
|  | TypeName | Gets the name of the this edge type |
|  | Unrestricted | Is this edge type restricted to a certain head and tail vertex type? |



Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|   | Compare | Compares two edge types by id |
|  | CompareTo | Compares two EdgeType objects by id |
|  | CountEdges | Get a count of existing edges of this type |
|  | Equals | Compares two edge types by id |
|  | FindProperty | Gets the property type given a property type name |
|  | GetEdge(Int32, Boolean, Boolean) | Gets an edge given an edge id. Throws if no such edge exist. |
|  | GetEdge(Int32, Vertex, Vertex) | Get an edge for a given id |
|  | GetEdges | Enumerates all edges of this type |
|  | GetHashCode | Get a hash code for an edge type based on type id |
|  | GetPropertyKeys | Return all the keys associated with the edge type. |
|  | GetPropertyTypes | Return all the property types associated with edge type. |
|  | GetPropertyValue | Gets a property value given an edge id and a property type |
|  | NewEdge | Create an edge between tail and head vertex |
|  | NewEdgeX | Not yet implemented |
|  | NewProperty | Creates a new Property. |
|  | Remove | Removes this EdgeType from a graph. An exception is thrown if the EdgeType is in use. |
|  | RemoveEdge | Removing an edge from this edge type |

VelocityDB Class Library

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetPropertyValue(EdgeType, Int32, PropertyType, IComparable) | Sets an edge property given an edge id, property type and a value |
|  SetPropertyValue(VertexType, Int32, PropertyType, IComparable) | |
|  ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[VelocityGraph Namespace](#)

EdgeType Constructor

Creates a new edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType(  
    int aTypeId,  
    string aTypeName,  
    VertexType tailType,  
    VertexType headType,  
    bool birectional,  
    EdgeType baseType,  
    Graph graph  
)
```

VB

```
Public Sub New (  
    aTypeId As Integer,  
    aTypeName As String,  
    tailType As VertexType,  
    headType As VertexType,  
    birectional As Boolean,  
    baseType As EdgeType,  
    graph As Graph  
)
```

C++

```
public:  
EdgeType (  
    int aTypeId,  
    String^ aTypeName,  
    VertexType^ tailType,  
    VertexType^ headType,  
    bool birectional,  
    EdgeType^ baseType,  
    Graph^ graph  
)
```

F#

```
new :  
    aTypeId : int *  
    aTypeName : string *  
    tailType : VertexType *  
    headType : VertexType *  
    birectional : bool *  
    baseType : EdgeType *  
    graph : Graph -> EdgeType
```

Parameters

aTypeId

Type: [System.Int32](#)

The id to use for the new edge type

aTypeName

Type: [System.String](#)

A type name to use

tailType

Type: [VelocityGraph.VertexType](#)

Restrict tail vertex to a certain vertex type

headType

Type: [VelocityGraph.VertexType](#)

Restrict head vertex to a certain vertex type

bidirectional

Type: [System.Boolean](#)

Is this edge type bidirectional (going both ways)

baseType

Type: [VelocityGraph.EdgeType](#)

A base type can be specified

graph

Type: [VelocityGraph.Graph](#)

The owning graph

See Also









[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.EdgeType Properties

The [EdgeType](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------------|
|  | Directed | Is Directed if edge type is not birectionalal |
|  | HeadType | Gets the Head VertexType of the edge (might not be set) |
|  | MyGraph | Graph for which this EdgeType belongs to |
|  | SubTypes | Sub types of this EdgeType |
|  | TailType | Gets the Tail VertexType of the edge (might not be set) |
|  | TypeId | Gets the id of the edge type |
|  | TypeName | Gets the name of the this edge type |
|  | Unrestricted | Is this edge type restricted to a certain head and tail vertex type? |

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Directed Property

Is Directed if edge type is not birectionalal

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Directed { get; }
```

VB

```
Public ReadOnly Property Directed As Boolean  
    Get
```

C++

```
public:  
property bool Directed {  
    bool get ();  
}
```

F#

```
member Directed : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.HeadType Property

Gets the Head VertexType of the edge (might not be set)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType HeadType { get; }
```

VB

```
Public ReadOnly Property HeadType As VertexType  
    Get
```

C++

```
public:  
property VertexType^ HeadType {  
    VertexType^ get ();  
}
```

F#

```
member HeadType : VertexType with get
```

Property Value

Type: [VertexType](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.MyGraph Property [Graph](#)

for which this [EdgeType](#) belongs to

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Graph MyGraph { get; }
```

VB

```
Public ReadOnly Property MyGraph As Graph  
    Get
```

C++

```
public:  
property Graph^ MyGraph {  
    Graph^ get ();  
}
```

F#

```
member MyGraph : Graph with get
```

Property Value

Type: [Graph](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.SubTypes Property

Sub types of this [EdgeType](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<EdgeType> SubTypes { get; }
```

VB

```
Public ReadOnly Property SubTypes As List(Of EdgeType)  
    Get
```

C++

```
public:  
property List<EdgeType^>^ SubTypes {  
    List<EdgeType^>^ get ();  
}
```

F#

```
member SubTypes : List<EdgeType> with get
```

Property Value

Type: [List\(EdgeType\)](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.TailType Property

Gets the Tail VertexType of the edge (might not be set)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType TailType { get; }
```

VB

```
Public ReadOnly Property TailType As VertexType  
    Get
```

C++

```
public:  
property VertexType^ TailType {  
    VertexType^ get ();  
}
```

F#

```
member TailType : VertexType with get
```

Property Value

Type: [VertexType](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.TypeId Property

Gets the id of the edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int TypeId { get; }
```

VB

```
Public ReadOnly Property TypeId As Integer  
    Get
```

C++

```
public:  
property int TypeId {  
    int get ();  
}
```

F#

```
member TypeId : int with get
```

Property Value

Type: [Int32](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.TypeName Property

Gets the name of the this edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TypeName { get; }
```

VB

```
Public ReadOnly Property TypeName As String  
    Get
```

C++

```
public:  
property String^ TypeName {  
    String^ get ();  
}
```

F#

```
member TypeName : string with get
```

Property Value

Type: [String](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Unrestricted Property

Is this edge type restricted to a certain head and tail vertex type?

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Unrestricted { get; }
```

VB

```
Public ReadOnly Property Unrestricted As Boolean  
    Get
```

C++

```
public:  
property bool Unrestricted {  
    bool get ();  
}
```

F#

```
member Unrestricted : bool with get
```

Property Value

Type: [Boolean](#)

See Also






















[EdgeType Class](#)

[VelocityGraph Namespace](#)



EdgeType.EdgeType Methods

The [EdgeType](#) type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Compare | Compares two edge types by id |
|  CompareTo | Compares two EdgeType objects by id |
|  CountEdges | Get a count of existing edges of this type |
|  Equals | Compares two edge types by id |
|  FindProperty | Gets the property type given a property type name |
|  GetEdge(Int32, Boolean, Boolean) | Gets an edge given an edge id. Throws if no such edge exist. |
|  GetEdge(Int32, Vertex, Vertex) | Get an edge for a given id |
|  GetEdges | Enumerates all edges of this type |
|  GetHashCode | Get a hash code for an edge type based on type id |
|  GetPropertyKeys | Return all the keys associated with the edge type. |
|  GetPropertyTypes | Return all the property types associated with edge type. |
|  GetPropertyValue | Gets a property value given an edge id and a property type |
|  NewEdge | Create an edge between tail and head vertex |
|  NewEdgeX | Not yet implemented |
|  NewProperty | Creates a new Property. |
|  Remove | Removes this EdgeType from a graph. An exception is thrown if the EdgeType is in use. |
|  RemoveEdge | Removing an edge from this edge type |
|  SetPropertyValue(EdgeType, Int32, PropertyType, IComparable) | Sets an edge property given an edge id, property type and a value |
|  SetPropertyValue(VertexType, Int32, PropertyType, IComparable) | |
|  ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Compare Method

Compares two edge types by id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static int Compare(  
    EdgeType aId,  
    EdgeType bId  
)
```

VB

```
Public Shared Function Compare (  
    aId As EdgeType,  
    bId As EdgeType  
) As Integer
```

C++

```
public:  
static int Compare(  
    EdgeType^ aId,  
    EdgeType^ bId  
)
```

F#

```
static member Compare :  
    aId : EdgeType *  
    bId : EdgeType -> int
```

Parameters

aId

Type: [VelocityGraph.EdgeType](#)

edge type 1

bId

Type: [VelocityGraph.EdgeType](#)

edge type 2

Return Value

Type: [Int32](#)

0 if edge types are equal, -1 if aId is less than bId; otherwise 1

See Also

[EdgeType Class](#)

VelocityDB Class Library

[VelocityGraph Namespace](#)

EdgeType.CompareTo Method

Compares two EdgeType objects by id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int CompareTo(  
    EdgeType obj  
)
```

VB

```
Public Function CompareTo (  
    obj As EdgeType  
) As Integer
```

C++

```
public:  
virtual int CompareTo(  
    EdgeType^ obj  
) sealed
```

F#

```
abstract CompareTo :  
    obj : EdgeType -> int  
override CompareTo :  
    obj : EdgeType -> int
```

Parameters

obj

Type: [VelocityGraph.EdgeType](#)

The object to compare with

Return Value

Type: [Int32](#)

a negative number if less, 0 if equal or else a positive number

Implements

[IComparable\(T\).CompareTo\(T\)](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.CountEdges Method

Get a count of existing edges of this type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long CountEdges ()
```

VB

```
Public Function CountEdges As Long
```

C++

```
public:  
long long CountEdges ()
```

F#

```
member CountEdges : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.EdgeType.CountEdges"]

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Equals Method

Compares two edge types by id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    EdgeType x,  
    EdgeType y  
)
```

VB

```
Public Function Equals (  
    x As EdgeType,  
    y As EdgeType  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    EdgeType^ x,  
    EdgeType^ y  
) sealed
```

F#

```
abstract Equals :  
    x : EdgeType *  
    y : EdgeType -> bool  
override Equals :  
    x : EdgeType *  
    y : EdgeType -> bool
```

Parameters

x

Type: [VelocityGraph.EdgeType](#)

edge type 1

y

Type: [VelocityGraph.EdgeType](#)

edge type 2

Return Value

Type: [Boolean](#)

true if the specified objects are equal; otherwise, false.

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.FindProperty Method

Gets the property type given a property type name

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType FindProperty(  
    string name,  
    bool doBaseType = true  
)
```

VB

```
Public Function FindProperty (  
    name As String,  
    Optional doBaseType As Boolean = true  
) As PropertyType
```

C++

```
public:  
PropertyType^ FindProperty(  
    String^ name,  
    bool doBaseType = true  
)
```

F#

```
member FindProperty :  
    name : string *  
    ?doBaseType : bool  
(* Defaults:  
    let _doBaseType = defaultArg doBaseType true  
)  
-> PropertyType
```

Parameters

name

Type: [System.String](#)

a property type name

doBaseType (Optional)

Type: [System.Boolean](#)

[Missing <param name="doBaseType"/> documentation for "M:VelocityGraph.EdgeType.FindProperty(System.String,System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [PropertyType](#)

a looked up property type or null if no such property type exist



See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetEdge Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------|
|  | GetEdge(Int32, Boolean, Boolean) | Gets an edge given an edge id. Throws if no such edge exist. |
|  | GetEdge(Int32, Vertex, Vertex) | Get an edge for a given id |

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetEdge Method (Int32, Boolean, Boolean)

Gets an edge given an edge id. Throws if no such edge exist.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge GetEdge(  
    int edgeId,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Function GetEdge (  
    edgeId As Integer,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Edge
```

C++

```
public:  
Edge^ GetEdge(  
    int edgeId,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

F#

```
member GetEdge :  
    edgeId : int *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let _polymorphic = defaultArg polymorphic false  
    let _errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Edge
```

Parameters

edgeId

Type: [System.Int32](#)

The id of the edge

polymorphic (Optional)

Type: [System.Boolean](#)

VelocityDB Class Library

If true and id isn't found in this EdgeType continue search into sub types

errorIfNotFound (Optional)

Type: [System.Boolean](#)

Indicate what to do if [Edge](#) does not exist

Return Value

Type: [Edge](#)

The edge with matching id if it exists

See Also

[EdgeType Class](#)

[GetEdge Overload](#)

[VelocityGraph Namespace](#)

EdgeType.GetEdge Method (Int32, Vertex, Vertex)

Get an edge for a given id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge GetEdge(  
    int edgeId,  
    Vertex tailVertex,  
    Vertex headVertex  
)
```

VB

```
Public Function GetEdge (  
    edgeId As Integer,  
    tailVertex As Vertex,  
    headVertex As Vertex  
) As Edge
```

C++

```
public:  
Edge^ GetEdge(  
    int edgeId,  
    Vertex^ tailVertex,  
    Vertex^ headVertex  
)
```

F#

```
member GetEdge :  
    edgeId : int *  
    tailVertex : Vertex *  
    headVertex : Vertex -> Edge
```

Parameters

edgeId

Type: [System.Int32](#)

Id of edge

tailVertex

Type: [VelocityGraph.Vertex](#)

the tail vertex of the edge

headVertex

Type: [VelocityGraph.Vertex](#)

the head vertex of the edge

Return Value

Type: [Edge](#)

[Missing <returns> documentation for

"M:VelocityGraph.EdgeType.GetEdge(System.Int32,VelocityGraph.Vertex,VelocityGraph.Vertex)"]

See Also

[EdgeType Class](#)

[GetEdge Overload](#)

[VelocityGraph Namespace](#)

EdgeType.GetEdges Method

Enumerates all edges of this type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Edge> GetEdges (  
    bool polymorphic = false  
)
```

VB

```
Public Function GetEdges (  
    Optional polymorphic As Boolean = false  
) As IEnumerable(Of Edge)
```

C++

```
public:  
IEnumerable<Edge^>^ GetEdges (  
    bool polymorphic = false  
)
```

F#

```
member GetEdges :  
    ?polymorphic : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
*)  
-> IEnumerable<Edge>
```

Parameters

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also include all edges of sub types

Return Value

Type: [IEnumerable\(Edge\)](#)

Enumeration of edges of this type

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetHashCode Method

Get a hash code for an edge type based on type id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode(  
    EdgeType t  
)
```

VB

```
Public Function GetHashCode (  
    t As EdgeType  
) As Integer
```

C++

```
public:  
virtual int GetHashCode(  
    EdgeType^ t  
) sealed
```

F#

```
abstract GetHashCode :  
    t : EdgeType -> int  
override GetHashCode :  
    t : EdgeType -> int
```

Parameters

t

Type: [VelocityGraph.EdgeType](#)

edge type to get hash code for

Return Value

Type: [Int32](#)

Hash code of an edge type

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetPropertyKeys Method

Return all the keys associated with the edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
IEnumerable<String^>^ GetPropertyKeys ()
```

F#

```
member GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

the set of all string keys associated with the edge type

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetPropertyTypes Method

Return all the property types associated with edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<PropertyType> GetPropertyTypes ()
```

VB

```
Public Function GetPropertyTypes As IEnumerable (Of PropertyType)
```

C++

```
public:  
IEnumerable<PropertyType^> GetPropertyTypes ()
```

F#

```
member GetPropertyTypes : unit -> IEnumerable<PropertyType>
```

Return Value

Type: [IEnumerable\(PropertyType\)](#)

the set of property types associated with the edge type

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.GetPropertyValue Method

Gets a property value given an edge id and a property type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IComparable GetPropertyValue(  
    int elementId,  
    PropertyType property  
)
```

VB

```
Public Function GetPropertyValue (  
    elementId As Integer,  
    property As PropertyType  
) As IComparable
```

C++

```
public:  
IComparable^ GetPropertyValue(  
    int elementId,  
    PropertyType^ property  
)
```

F#

```
member GetPropertyValue :  
    elementId : int *  
    property : PropertyType -> IComparable
```

Parameters

elementId

Type: [System.Int32](#)

an edge id

property

Type: [VelocityGraph.PropertyType](#)

a property type

Return Value

Type: [IComparable](#)

a property value

See Also

[EdgeType Class](#)

VelocityDB Class Library

[VelocityGraph Namespace](#)

EdgeType.NewEdge Method

Create an edge between tail and head vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge NewEdge(  
    Vertex tail,  
    Vertex head  
)
```

VB

```
Public Function NewEdge (  
    tail As Vertex,  
    head As Vertex  
) As Edge
```

C++

```
public:  
Edge^ NewEdge (  
    Vertex^ tail,  
    Vertex^ head  
)
```

F#

```
member NewEdge :  
    tail : Vertex *  
    head : Vertex -> Edge
```

Parameters

tail

Type: [VelocityGraph.Vertex](#)

selected tail vertex

head

Type: [VelocityGraph.Vertex](#)

selected head vertex

Return Value

Type: [Edge](#)

a new edge

See Also

[EdgeType Class](#)

VelocityDB Class Library

[VelocityGraph Namespace](#)

EdgeType.NewEdgeX Method

Not yet implemented

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge NewEdgeX(  
    WeakReferenceList<PropertyType> propertyType,  
    PropertyType tailAttr,  
    Object tailV,  
    PropertyType headAttr,  
    Object headV,  
    SessionBase session  
)
```

VB

```
Public Function NewEdgeX (  
    propertyType As WeakReferenceList(Of PropertyType),  
    tailAttr As PropertyType,  
    tailV As Object,  
    headAttr As PropertyType,  
    headV As Object,  
    session As SessionBase  
) As Edge
```

C++

```
public:  
Edge^ NewEdgeX(  
    WeakReferenceList<PropertyType^>^ propertyType,  
    PropertyType^ tailAttr,  
    Object^ tailV,  
    PropertyType^ headAttr,  
    Object^ headV,  
    SessionBase^ session  
)
```

F#

```
member NewEdgeX :  
    propertyType : WeakReferenceList<PropertyType> *  
    tailAttr : PropertyType *  
    tailV : Object *  
    headAttr : PropertyType *  
    headV : Object *  
    session : SessionBase -> Edge
```

Parameters

propertyType

Type: [VelocityDb.Collection.WeakReferenceList\(PropertyType\)](#)

[Missing <param name="propertyType"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

tailAttr

Type: [VelocityGraph.PropertyType](#)

[Missing <param name="tailAttr"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

tailV

Type: [System.Object](#)

[Missing <param name="tailV"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

headAttr

Type: [VelocityGraph.PropertyType](#)

[Missing <param name="headAttr"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

headV

Type: [System.Object](#)

[Missing <param name="headV"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

session

Type: [VelocityDb.Session.SessionBase](#)

[Missing <param name="session"/> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.PropertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

Return Value

Type: [Edge](#)

[Missing <returns> documentation for "M:VelocityGraph.EdgeType.NewEdgeX(VelocityDb.Collection.WeakReferenceList{VelocityGraph.Prop

ertyType},VelocityGraph.PropertyType,System.Object,VelocityGraph.PropertyType,System.Object,VelocityDb.Session.SessionBase)"]

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.NewProperty Method

Creates a new Property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType NewProperty(  
    string name,  
    DataType dt,  
    PropertyKind kind  
)
```

VB

```
Public Function NewProperty (  
    name As String,  
    dt As DataType,  
    kind As PropertyKind  
) As PropertyType
```

C++

```
public:  
PropertyType^ NewProperty(  
    String^ name,  
    DataType dt,  
    PropertyKind kind  
)
```

F#

```
member NewProperty :  
    name : string *  
    dt : DataType *  
    kind : PropertyKind -> PropertyType
```

Parameters

name

Type: [System.String](#)

Unique name for the new Property.

dt

Type: [VelocityGraph.DataType](#)

Data type for the new Property.

kind

Type: [VelocityGraph.PropertyKind](#)

Property kind.

VelocityDB Class Library

Return Value

Type: [PropertyType](#)

a Property.

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Remove Method

Removes this [EdgeType](#) from a graph. An exception is thrown if the [EdgeType](#) is in use.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove ()
```

VB

```
Public Sub Remove
```

C++

```
public:  
void Remove ()
```

F#

```
member Remove : unit -> unit
```

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.RemoveEdge Method

Removing an edge from this edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (  
    Edge edge  
)
```

VB

```
Public Sub RemoveEdge (  
    edge As Edge  
)
```

C++

```
public:  
void RemoveEdge (  
    Edge^ edge  
)
```

F#

```
member RemoveEdge :  
    edge : Edge -> unit
```

Parameters

edge

Type: [VelocityGraph.Edge](#)

an edge to remove



See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.SetPropertyValue Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------|-------------------------------------------------------------------|
|  | SetPropertyValue(EdgeType, Int32, PropertyType, IComparable) | Sets an edge property given an edge id, property type and a value |
|  | SetPropertyValue(VertexType, Int32, PropertyType, IComparable) | |

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.SetPropertyValue Method (EdgeType, Int32, PropertyType, IComparable)

Sets an edge property given an edge id, property type and a value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetPropertyValue (
    EdgeType et,
    int elementId,
    PropertyType property,
    IComparable v
)
```

VB

```
Public Sub SetPropertyValue (
    et As EdgeType,
    elementId As Integer,
    property As PropertyType,
    v As IComparable
)
```

C++

```
public:
void SetPropertyValue (
    EdgeType^ et,
    int elementId,
    PropertyType^ property,
    IComparable^ v
)
```

F#

```
member SetPropertyValue :
    et : EdgeType *
    elementId : int *
    property : PropertyType *
    v : IComparable -> unit
```

Parameters

et

Type: [VelocityGraph.EdgeType](#)

[Missing <param name="et"/> documentation for

"M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,VelocityGraph.PropertyType,System.IComparable)"]

VelocityDB Class Library

elementId

Type: [System.Int32](#)

an edge id

property

Type: [VelocityGraph.PropertyType](#)

a property type

v

Type: [System.IComparable](#)

a value

See Also

[EdgeType Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

EdgeType.SetPropertyValue Method (VertexType, Int32, PropertyType, IComparable)

[Missing <summary> documentation for "M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph.PropertyType,System.IComparable)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetPropertyValue(  
    VertexType vt,  
    int elementId,  
    PropertyType property,  
    IComparable v  
)
```

VB

```
Public Sub SetPropertyValue (  
    vt As VertexType,  
    elementId As Integer,  
    property As PropertyType,  
    v As IComparable  
)
```

C++

```
public:  
void SetPropertyValue(  
    VertexType^ vt,  
    int elementId,  
    PropertyType^ property,  
    IComparable^ v  
)
```

F#

```
member SetPropertyValue :  
    vt : VertexType *  
    elementId : int *  
    property : PropertyType *  
    v : IComparable -> unit
```

Parameters

vt

Type: [VelocityGraph.VertexType](#)

[Missing <param name="vt"/> documentation for
"M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph
.PropertyType,System.IComparable)"]

elementId

Type: [System.Int32](#)

[Missing <param name="elementId"/> documentation for
"M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph
.PropertyType,System.IComparable)"]

property

Type: [VelocityGraph.PropertyType](#)

[Missing <param name="property"/> documentation for
"M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph
.PropertyType,System.IComparable)"]

v

Type: [System.IComparable](#)

[Missing <param name="v"/> documentation for
"M:VelocityGraph.EdgeType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph
.PropertyType,System.IComparable)"]

See Also

[EdgeType Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

EdgeType.ToString Method

Displays class name plus object id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

EdgeType.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[EdgeType Class](#)

[VelocityGraph Namespace](#)

Element Class

Base class for Edge and Vertex

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Element

[VelocityGraph.Edge](#)

[VelocityGraph.Vertex](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class Element : DictionaryElement
```

VB

```
Public MustInherit Class Element  
    Inherits DictionaryElement
```

C++

```
public ref class Element abstract : public DictionaryElement
```

F#

```
[<AbstractClassAttribute>]  
type Element =  
    class  
        inherit DictionaryElement  
    end
```

The **Element** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------------------|
|  | GetHashCode | Use id as hash code (Overrides Object.GetHashCode() .) |


See Also

[VelocityGraph Namespace](#)

Element.Element Methods

The [Element](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------------------|
|  | GetHashCode | Use id as hash code (Overrides Object.GetHashCode() .) |

See Also

[Element Class](#)

[VelocityGraph Namespace](#)

Element.GetHashCode Method

Use id as hash code

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

The hash code given by id

See Also

[Element Class](#)

[VelocityGraph Namespace](#)

Graph Class

Graph is the root object of a graph. Most graph api is on this class but useful api also exist on [VertexType](#) and [EdgeType](#).

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityGraph.Graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class Graph : OptimizedPersistable, IGraph
```

VB

```
<SerializableAttribute>
Public Class Graph
    Inherits OptimizedPersistable
    Implements IGraph
```

C++


```
[SerializableAttribute]
public ref class Graph : public OptimizedPersistable,
    IGraph
```

F#




```
[<SerializableAttribute>]
type Graph =
    class
        inherit OptimizedPersistable
        interface IGraph
    end
```

The **Graph** type exposes the following members.
















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|----------------------------|
|  | Graph | Creates a new Graph |




















Properties


| Name | Description |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
|  Features | Returns the features of the underlying graph. |
|  ObjectsPerPage | Place this type of object on its own page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  VertexIdSetPerType | <code>true</code> if using a vertex id set per type. Decided by constructor parameter when creating a Graph |

Methods
















| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation. |
|  AddVertex() | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |
|  AddVertex(Object) | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |
|  Clear | Removes all edges and vertices from the graph. |
|  CombineIntersection | Combines to sets of edges by intersection |
|  ContainsVertex | Returns a value indicating whether a Vertex exists for the specified vertex id. |
|  CountEdges | Returns a count of all edges in this graph |
|  CountVertices | Get a count of all vertices in this graph |
|  Edges | Enumerates all the edges of the given type between two given nodes (tail and head). |
|  ExportToGraphJson(Stream) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(String) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(String, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphSon(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a GraphSon JSON OutputStream. |
|  ExportToGraphSon(String, IEnumerable(String), | Write the data in a Graph to a GraphSon JSON OutputStream. |

| | | |
|---|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| | IEnumerable(String), GraphSonMode) | |
| ⇒ | FindEdgeType | Finds the type id associated with a particular edge type. Lookup by name. |
| ⇒ | FindEdgeTypes | Get an array of all edge types in this graph |
| ⇒ | FindVertex | Finds vertex having the given value for the given property. |
| ⇒ | FindVertexProperty | Finds a PropertyType of VertexType . |
| ⇒ | FindVertexType | Finds the type id associated with a particular vertex type. Lookup by name. |
| ⇒ | FindVertexTypes | Get an array of all vertex types in this graph |
| ⇒ | GetEdge | Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null. |
| ⇒ | GetEdges() | Return an iterable to all the edges in the graph. |
| ⇒ | GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. |
| ⇒ | GetEdges(T)(String, T) | Return an iterable to all the edges in the graph that have a particular key/value property. |
| ⇒ | GetEdgeType | Get a EdgeType associated with an Edge id |
| ⇒ | GetValues | Enumerates all the values for the given vertex/edge property |
| ⇒ | GetVertex(Int32) | Look up a Vertex |
| ⇒ | GetVertex(Object) | Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null. |
| ⇒ | GetVertexType | Given a Vertex id, returns the corresponding Vertex . |
| ⇒ | GetVertices() | Return an iterable to all the vertices in the graph. |
| ⇒ | GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. |
| ⇒ | ImportGraphJson(String) | Input the GraphJson JSON stream data into the graph. In practice, usually the provided graph is empty. |
| ⇒ | ImportGraphJson(Stream, Int32) | Input the GraphJson JSON stream data into the graph. In practice, usually the provided graph is empty. |
| ⇒ | ImportGraphSon(String) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |
| ⇒ | ImportGraphSon(Stream, Int32) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |
| ⇒ | NewEdge(EdgeType, Vertex, Vertex) | Creates a new edge instance. |
| ⇒ | NewEdge(EdgeType, PropertyType, Object, PropertyType, Object) | Creates a new edge instance. The tail of the edge will be any node having the given tailV Value for the given tailAttr Property |

| | | |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | identifier, and the head of the edge will be any node having the given headV Value for the given headAttr Property identifier. |
|  | NewEdgeProperty | Creates a new Property. |
|  | NewEdgeType(String, Boolean, EdgeType) | Creates a new edge type. |
|  | NewEdgeType(String, Boolean, VertexType, VertexType, EdgeType) | Creates a new edge type. |
|  | NewVertex | Creates a new node instance. |
|  | NewVertexProperty | Creates a new Property. |
|  | NewVertexType | Creates a new node type. |
|  | Open | Opens a graph |
|  | | |
|  | Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |
|  | Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph. |
|  | RemoveEdge | Remove the provided edge from the graph. |
|  | RemovePropertyType | Removes a property type except if any vertex or edge is using it |
|  | RemoveVertex | Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well. |
|  | RemoveVertexType | Removes a VertexType from this graph. An exception is thrown if the VertexType is in use. |
|  | Select(PropertyType, Func(IComparable, IComparable, Boolean), IComparable) | Enumerates all elements satisfying the given condition for the given property and value |
|  | Select(PropertyType, Func(IComparable, IComparable, IComparable, Boolean), IComparable, IComparable) | numerates all elements satisfying the given condition for the given property and value range |
|  | Shutdown | A shutdown function is required to properly close the graph. This is important for implementations that utilize disk-based serializations. |
|  | ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |
|  | Traverse | Selects all neighbor Vertices from or to each of the node OID in the given collection and for the given edge type. |

| | |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

VelocityDB Class Library

See Also

[VelocityGraph Namespace](#)

Graph Constructor

Creates a new [Graph](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Graph(  
    SessionBase session,  
    bool vertexIdSetPerVertexType = true  
)
```

VB

```
Public Sub New (  
    session As SessionBase,  
    Optional vertexIdSetPerVertexType As Boolean = true  
)
```

C++

```
public:  
Graph(  
    SessionBase^ session,  
    bool vertexIdSetPerVertexType = true  
)
```

F#

```
new :  
    session : SessionBase *  
    ?vertexIdSetPerVertexType : bool  
(* Defaults:  
    let _vertexIdSetPerVertexType = defaultArg vertexIdSetPerVertexType  
true  
*)  
-> Graph
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The active session

vertexIdSetPerVertexType (Optional)

Type: [System.Boolean](#)

Set to *false* (False in Visual Basic) if you want graph wide unique [Vertex](#) ids, by default *true* (True in Visual Basic) each [VertexType](#) maintains its own set of [Vertex](#) ids

VelocityDB Class Library

See Also




[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Graph Properties

The [Graph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|  | Features | Returns the features of the underlying graph. |
|  | ObjectsPerPage | Place this type of object on its own page (Overrides OptimizedPersistable.ObjectsPerPage.) |
|  | VertexIdSetPerType | true if using a vertex id set per type. Decided by constructor parameter when creating a Graph |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Features Property

Returns the features of the underlying graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Features Features { get; }
```

VB

```
Public Overridable ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get ();  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ObjectsPerPage Property

Place this type of object on its own page

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ushort ObjectsPerPage { get; }
```

VB

```
Public Overrides ReadOnly Property ObjectsPerPage As UShort  
    Get
```

C++

```
public:  
virtual property unsigned short ObjectsPerPage {  
    unsigned short get () override;  
}
```

F#

```
abstract ObjectsPerPage : uint16 with get  
override ObjectsPerPage : uint16 with get
```

Return Value

Type: [UInt16](#)

The default maximum number of objects per page

Implements

[IOptimizedPersistable.ObjectsPerPage](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.VertexIdSetPerType Property

true

if using a vertex id set per type. Decided by constructor parameter when creating a [Graph](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool VertexIdSetPerType { get; }
```

VB

```
Public ReadOnly Property VertexIdSetPerType As Boolean  
    Get
```

C++

```
public:  
property bool VertexIdSetPerType {  
    bool get ();  
}
```

F#

```
member VertexIdSetPerType : bool with get
```

Property Value

Type: [Boolean](#)

See Also


















[Graph Class](#)










[VelocityGraph Namespace](#)


















Graph.Graph Methods

The [Graph](#) type exposes the following members.
















Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation. |
|  AddVertex() | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |
|  AddVertex(Object) | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |
|  Clear | Removes all edges and vertices from the graph. |
|  CombineIntersection | Combines to sets of edges by intersection |
|  ContainsVertex | Returns a value indicating whether a Vertex exists for the specified vertex id. |
|  CountEdges | Returns a count of all edges in this graph |
|  CountVertices | Get a count of all vertices in this graph |
|  Edges | Enumerates all the edges of the given type between two given nodes (tail and head). |
|  ExportToGraphJson(Stream) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(String) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphJson(String, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  ExportToGraphSon(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a GraphSon JSON OutputStream. |
|  ExportToGraphSon(String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a GraphSon JSON OutputStream. |
|  FindEdgeType | Finds the type id associated with a particular edge type. Lookup by name. |
|  FindEdgeTypes | Get an array of all edge types in this graph |
|  FindVertex | Finds vertex having the given value for the given property. |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  FindVertexProperty | Finds a PropertyType of VertexType . |
|  FindVertexType | Finds the type id associated with a particular vertexe type. Lookup by name. |
|  FindVertexTypes | Get an array of all vertex types in this graph |
|  GetEdge | Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null. |
|  GetEdges() | Return an iterable to all the edges in the graph. |
|  GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. |
|  GetEdges(T)(String, T) | Return an iterable to all the edges in the graph that have a particular key/value property. |
|  GetEdgeType | Get a EdgeType associated with an Edge id |
|  GetValues | Enumerates all the values for the given vertex/edge property |
|  GetVertex(Int32) | Look up a Vertex |
|  GetVertex(Object) | Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null. |
|  GetVertexType | Given a Vertex id, returns the corresponding Vertex . |
|  GetVertices() | Return an iterable to all the vertices in the graph. |
|  GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. |
|  ImportGraphJson(String) | Input the GraphJson JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  ImportGraphJson(Stream, Int32) | Input the GrapJson JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  ImportGraphSon(String) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  ImportGraphSon(Stream, Int32) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  NewEdge(EdgeType, Vertex, Vertex) | Creates a new edge instance. |
|  NewEdge(EdgeType, PropertyType, Object, PropertyType, Object) | Creates a new edge instance. The tail of the edge will be any node having the given tailV Value for the given tailAttr Property identifier, and the head of the edge will be any node having the given headV Value for the given headAttr Property identifier. |
|  NewEdgeProperty | Creates a new Property. |
|  NewEdgeType(String, Boolean, EdgeType) | Creates a new edge type. |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  NewEdgeType(String, Boolean, VertexType, VertexType, EdgeType) | Creates a new edge type. |
|  NewVertex | Creates a new node instance. |
|  NewVertexProperty | Creates a new Property. |
|  NewVertexType | Creates a new node type. |
|  Open | Opens a graph |
|  Persist | Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way. (Overrides OptimizedPersistable.Persist(Placement, SessionBase, Boolean, Boolean, Queue(IOptimizedPersistable)) .) |
|  Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph. |
|  RemoveEdge | Remove the provided edge from the graph. |
|  RemovePropertyType | Removes a property type except if any vertex or edge is using it |
|  RemoveVertex | Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well. |
|  RemoveVertexType | Removes a VertexType from this graph. An exception is thrown if the VertexType is in use. |
|  Select(PropertyType, Func(IComparable, IComparable, Boolean), IComparable) | Enumerates all elements satisfying the given condition for the given property and value |
|  Select(PropertyType, Func(IComparable, IComparable, IComparable, Boolean), IComparable, IComparable) | numerates all elements satisfying the given condition for the given property and value range |
|  Shutdown | A shutdown function is required to properly close the graph. This is important for implementations that utilize disk-based serializations. |
|  ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |
|  Traverse | Selects all neighbor Vertices from or to each of the node OID in the given collection and for the given edge type. |
|  Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  | CreateTinkerGraph | (Defined by GraphHelpers.) |
|  | GraphString | (Defined by StringFactory.) |
|  | LoadGml | (Defined by GraphHelpers.) |
|  | LoadGraphml | (Defined by GraphHelpers.) |
|  | LoadGraphson | (Defined by GraphHelpers.) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |

See Also

[Graph Class](#)[VelocityGraph Namespace](#)

Graph.AddEdge Method

Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Overridable Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
)
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

VelocityDB Class Library

Type: [System.Object](#)

the recommended object identifier

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex on the tail of the edge

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex on the head of the edge

label

Type: [System.String](#)

the label associated with the edge

Return Value

Type: [IEdge](#)

the newly created edge

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)



See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.AddVertex Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------|
|  | AddVertex() | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |
|  | AddVertex(Object) | Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.AddVertex Method

Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Vertex AddVertex()
```

VB

```
Public Overridable Function AddVertex As Vertex
```

C++

```
public:  
virtual Vertex^ AddVertex()
```

F#

```
abstract AddVertex : unit -> Vertex  
override AddVertex : unit -> Vertex
```

Return Value

Type: [Vertex](#)

the newly created vertex

See Also

[Graph Class](#)

[AddVertex Overload](#)

[VelocityGraph Namespace](#)

Graph.AddVertex Method (Object)

Create a new vertex (of VertexType "default"), add it to the graph, and return the newly created vertex.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Overridable Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
)
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

the recommended object identifier

Return Value

Type: [IVertex](#)

the newly created vertex

Implements

[IGraph.AddVertex\(Object\)](#)

See Also

[Graph Class](#)

[AddVertex Overload](#)

[VelocityGraph Namespace](#)

Graph.Clear Method

Removes all edges and vertices from the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Clear ()
```

VB

```
Public Sub Clear
```

C++

```
public:  
void Clear ()
```

F#

```
member Clear : unit -> unit
```

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.CombineIntersection Method

Combines to sets of edges by intersection

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static HashSet<Edge> CombineIntersection(  
    HashSet<Edge> objs1,  
    HashSet<Edge> objs2  
)
```

VB

```
Public Shared Function CombineIntersection (  
    objs1 As HashSet(Of Edge),  
    objs2 As HashSet(Of Edge)  
) As HashSet(Of Edge)
```

C++

```
public:  
static HashSet<Edge^>^ CombineIntersection(  
    HashSet<Edge^>^ objs1,  
    HashSet<Edge^>^ objs2  
)
```

F#

```
static member CombineIntersection :  
    objs1 : HashSet<Edge> *  
    objs2 : HashSet<Edge> -> HashSet<Edge>
```

Parameters

objs1

Type: [System.Collections.Generic.HashSet\(Edge\)](#)

first set of edges

objs2

Type: [System.Collections.Generic.HashSet\(Edge\)](#)

second set of edges

Return Value

Type: [HashSet\(Edge\)](#)

Intersection of sets

See Also

[Graph Class](#)

VelocityDB Class Library

[VelocityGraph Namespace](#)

Graph.ContainsVertex Method

Returns a value indicating whether a Vertex exists for the specified vertex id.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool ContainsVertex(  
    int vertexId  
)
```

VB

```
Public Function ContainsVertex (  
    vertexId As Integer  
) As Boolean
```

C++

```
public:  
bool ContainsVertex(  
    int vertexId  
)
```

F#

```
member ContainsVertex :  
    vertexId : int -> bool
```

Parameters

vertexId

Type: [System.Int32](#)

A Vertex id

Return Value

Type: [Boolean](#)

Returns `true` (`True` in Visual Basic) if a Vertex exist with specified *vertexId* is found in this [Graph](#); otherwise, `false` (`False` in Visual Basic).

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.CountEdges Method

Returns a count of all edges in this graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long CountEdges ()
```

VB

```
Public Function CountEdges As Long
```

C++

```
public:  
long long CountEdges ()
```

F#

```
member CountEdges : unit -> int64
```

Return Value

Type: [Int64](#)

the total count of edges in this graph

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.CountVertices Method

Get a count of all vertices in this graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long CountVertices ()
```

VB

```
Public Function CountVertices As Long
```

C++

```
public:  
long long CountVertices ()
```

F#

```
member CountVertices : unit -> int64
```

Return Value

Type: [Int64](#)

count of vertices

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Edges Method

Enumerates all the edges of the given type between two given nodes (tail and head).

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> Edges (  
    EdgeType etype,  
    Vertex tail,  
    Vertex head  
)
```

VB

```
Public Function Edges (  
    etype As EdgeType,  
    tail As Vertex,  
    head As Vertex  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
IEnumerable<IEdge^> Edges (  
    EdgeType^ etype,  
    Vertex^ tail,  
    Vertex^ head  
)
```

F#

```
member Edges :  
    etype : EdgeType *  
    tail : Vertex *  
    head : Vertex -> IEnumerable<IEdge>
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

Type of Edge

tail

Type: [VelocityGraph.Vertex](#)

Outgoing Vertex

head

Type: [VelocityGraph.Vertex](#)

Incoming Vertex

VelocityDB Class Library

Return Value

Type: [IEnumerable\(IEdge\)](#)

Enumeration of Edge





See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ExportToJson Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------|
|  | ExportToJson(Stream) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  | ExportToJson(String) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  | ExportToJson(Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |
|  | ExportToJson(String, GraphJsonSettings) | Write the data in a Graph to a GraphJson JSON OutputStream. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphJson Method (Stream)

Write the data in a Graph to a GraphJson JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExportToGraphJson (  
    Stream jsonOutputStream  
)
```

VB

```
Public Sub ExportToGraphJson (  
    jsonOutputStream As Stream  
)
```

C++

```
public:  
void ExportToGraphJson (  
    Stream^ jsonOutputStream  
)
```

F#

```
member ExportToGraphJson :  
    jsonOutputStream : Stream -> unit
```

Parameters

jsonOutputStream

Type: [System.IO.Stream](#)

the OutputStream to write to

See Also

[Graph Class](#)

[ExportToGraphJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ExportToJson Method (String)

Write the data in a Graph to a GraphJson JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public void ExportToJson(  
    string filename  
)
```

```
VB  
Public Sub ExportToJson (  
    filename As String  
)
```

```
C++  
public:  
void ExportToJson(  
    String^ filename  
)
```

```
F#  
member ExportToJson :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

the JSON file to write the Graph data to

See Also

[Graph Class](#)

[ExportToJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphJson Method (Stream, GraphJsonSettings)

Write the data in a Graph to a GraphJson JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExportToGraphJson(  
    Stream jsonOutputStream,  
    GraphJsonSettings settings  
)
```

VB

```
Public Sub ExportToGraphJson (  
    jsonOutputStream As Stream,  
    settings As GraphJsonSettings  
)
```

C++

```
public:  
void ExportToGraphJson (  
    Stream^ jsonOutputStream,  
    GraphJsonSettings^ settings  
)
```

F#

```
member ExportToGraphJson :  
    jsonOutputStream : Stream *  
    settings : GraphJsonSettings -> unit
```

Parameters

jsonOutputStream

Type: [System.IO.Stream](#)

the OutputStream to write to

settings

Type: [VelocityGraph.GraphJsonSettings](#)

Contains field names that the writer will use to map to BluePrints

See Also

[Graph Class](#)

[ExportToGraphJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphJson Method (String, GraphJsonSettings)

Write the data in a Graph to a GraphJson JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExportToGraphJson(  
    string filename,  
    GraphJsonSettings settings  
)
```

VB

```
Public Sub ExportToGraphJson (  
    filename As String,  
    settings As GraphJsonSettings  
)
```

C++

```
public:  
void ExportToGraphJson(  
    String^ filename,  
    GraphJsonSettings^ settings  
)
```

F#

```
member ExportToGraphJson :  
    filename : string *  
    settings : GraphJsonSettings -> unit
```

Parameters

filename

Type: [System.String](#)

the JSON file to write the Graph data to

settings

Type: [VelocityGraph.GraphJsonSettings](#)

Contains field names that the writer will use to map to BluePrints

See Also



[Graph Class](#)

[ExportToGraphJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphSon Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|  | ExportToGraphSon(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a GraphSon JSON OutputStream. |
|  | ExportToGraphSon(String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a GraphSon JSON OutputStream. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphSon Method (Stream, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a GraphSon JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExportToGraphSon (
    Stream jsonOutputStream,
    IEnumerable<string> vertexPropertyKeys = null,
    IEnumerable<string> edgePropertyKeys = null,
    GraphSonMode mode = GraphSonMode.NORMAL
)
```

VB

```
Public Sub ExportToGraphSon (
    jsonOutputStream As Stream,
    Optional vertexPropertyKeys As IEnumerable(Of String) = Nothing,
    Optional edgePropertyKeys As IEnumerable(Of String) = Nothing,
    Optional mode As GraphSonMode = GraphSonMode.NORMAL
)
```

C++

```
public:
void ExportToGraphSon (
    Stream^ jsonOutputStream,
    IEnumerable<String^>^ vertexPropertyKeys = nullptr,
    IEnumerable<String^>^ edgePropertyKeys = nullptr,
    GraphSonMode mode = GraphSonMode::NORMAL
)
```

F#

```
member ExportToGraphSon :
    jsonOutputStream : Stream *
    ?vertexPropertyKeys : IEnumerable<string> *
    ?edgePropertyKeys : IEnumerable<string> *
    ?mode : GraphSonMode
(* Defaults:
    let _vertexPropertyKeys = defaultArg vertexPropertyKeys null
    let _edgePropertyKeys = defaultArg edgePropertyKeys null
    let _mode = defaultArg mode GraphSonMode.NORMAL
*)
-> unit
```

VelocityDB Class Library

Parameters

jsonOutputStream

Type: [System.IO.Stream](#)

the JSON OutputStream to write the Graph data to

vertexPropertyKeys (Optional)

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the vertex elements to write to JSON

edgePropertyKeys (Optional)

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode (Optional)

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[Graph Class](#)

[ExportToGraphSon Overload](#)

[VelocityGraph Namespace](#)

Graph.ExportToGraphSon Method (String, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a GraphSon JSON OutputStream.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ExportToGraphSon (
    string filename,
    IEnumerable<string> vertexPropertyKeys = null,
    IEnumerable<string> edgePropertyKeys = null,
    GraphSonMode mode = GraphSonMode.NORMAL
)
```

VB

```
Public Sub ExportToGraphSon (
    filename As String,
    Optional vertexPropertyKeys As IEnumerable(Of String) = Nothing,
    Optional edgePropertyKeys As IEnumerable(Of String) = Nothing,
    Optional mode As GraphSonMode = GraphSonMode.NORMAL
)
```

C++

```
public:
void ExportToGraphSon (
    String^ filename,
    IEnumerable<String^>^ vertexPropertyKeys = nullptr,
    IEnumerable<String^>^ edgePropertyKeys = nullptr,
    GraphSonMode mode = GraphSonMode::NORMAL
)
```

F#

```
member ExportToGraphSon :
    filename : string *
    ?vertexPropertyKeys : IEnumerable<string> *
    ?edgePropertyKeys : IEnumerable<string> *
    ?mode : GraphSonMode
(* Defaults:
    let _vertexPropertyKeys = defaultArg vertexPropertyKeys null
    let _edgePropertyKeys = defaultArg edgePropertyKeys null
    let _mode = defaultArg mode GraphSonMode.NORMAL
*)
-> unit
```

VelocityDB Class Library

Parameters

filename

Type: [System.String](#)

the JSON file to write the Graph data to

vertexPropertyKeys (Optional)

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the vertex elements to write to JSON

edgePropertyKeys (Optional)

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode (Optional)

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[Graph Class](#)

[ExportToGraphSon Overload](#)

[VelocityGraph Namespace](#)

Graph.FindEdgeType Method

Finds the type id associated with a particular edge type. Lookup by name.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType FindEdgeType(  
    string name  
)
```

VB

```
Public Function FindEdgeType (  
    name As String  
) As EdgeType
```

C++

```
public:  
EdgeType^ FindEdgeType (  
    String^ name  
)
```

F#

```
member FindEdgeType :  
    name : string -> EdgeType
```

Parameters

name

Type: [System.String](#)

The name of the edge/node type being looked up

Return Value

Type: [EdgeType](#)

A node/edge type id or -1 if not found.

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.FindEdgeTypes Method

Get an array of all edge types in this graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<EdgeType> FindEdgeTypes ()
```

VB

```
Public Function FindEdgeTypes As IEnumerable (Of EdgeType)
```

C++

```
public:  
IEnumerable<EdgeType^> FindEdgeTypes ()
```

F#

```
member FindEdgeTypes : unit -> IEnumerable<EdgeType>
```

Return Value

Type: [IEnumerable\(EdgeType\)](#)

an array of edge types

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.FindVertex Method

Finds vertex having the given value for the given property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex FindVertex(  
    PropertyType property,  
    IComparable v,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Function FindVertex (  
    property As PropertyType,  
    v As IComparable,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
Vertex^ FindVertex(  
    PropertyType^ property,  
    IComparable^ v,  
    bool errorIfNotFound = true  
)
```

F#

```
member FindVertex :  
    property : PropertyType *  
    v : IComparable *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let_errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

v

Type: [System.IComparable](#)

a value to look for

errorIfNotFound (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

if `true`, throw an exception if an error is found

Return Value

Type: [Vertex](#)

the vertex matching

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.FindVertexProperty Method

Finds a [PropertyType](#) of [VertexType](#).

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType FindVertexProperty(  
    VertexType vertexType,  
    string name  
)
```

VB

```
Public Function FindVertexProperty (  
    vertexType As VertexType,  
    name As String  
) As PropertyType
```

C++

```
public:  
PropertyType^ FindVertexProperty(  
    VertexType^ vertexType,  
    String^ name  
)
```

F#

```
member FindVertexProperty :  
    vertexType : VertexType *  
    name : string -> PropertyType
```

Parameters

vertexType

Type: [VelocityGraph.VertexType](#)

vertex type with property type

name

Type: [System.String](#)

a name of a property

Return Value

Type: [PropertyType](#)

[Missing <returns> documentation for

"M:VelocityGraph.Graph.FindVertexProperty(VelocityGraph.VertexType,System.String)"]

VelocityDB Class Library

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.FindVertexType Method

Finds the type id associated with a particular vertexe type. Lookup by name.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType FindVertexType(  
    string name  
)
```

VB

```
Public Function FindVertexType (  
    name As String  
) As VertexType
```

C++

```
public:  
VertexType^ FindVertexType(  
    String^ name  
)
```

F#

```
member FindVertexType :  
    name : string -> VertexType
```

Parameters

name

Type: [System.String](#)

The name of the edge/node type being looked up

Return Value

Type: [VertexType](#)

A node/edge type id or -1 if not found.

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.FindVertexTypes Method

Get an array of all vertex types in this graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<VertexType> FindVertexTypes ()
```

VB

```
Public Function FindVertexTypes As IEnumerable (Of VertexType)
```

C++

```
public:  
IEnumerable<VertexType^> FindVertexTypes ()
```

F#

```
member FindVertexTypes : unit -> IEnumerable<VertexType>
```

Return Value

Type: [IEnumerable\(VertexType\)](#)

an array of vertex types

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetEdge Method

Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

the identifier of the edge to retrieved from the graph

Return Value

Type: [IEdge](#)

the edge referenced by the provided identifier or null when no such edge exists

Implements

[IGraph.GetEdge\(Object\)](#)




See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|---------------------------------------------------------------------------------------------|
|  | GetEdges() | Return an iterable to all the edges in the graph. |
|  | GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. |
|  | GetEdges(T)(String, T) | Return an iterable to all the edges in the graph that have a particular key/value property. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetEdges Method

Return an iterable to all the edges in the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable (Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

an iterable reference to all edges in the graph

Implements

[IGraph.GetEdges\(\)](#)

See Also

[Graph Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

Graph.GetEdges Method (String, Object)

Return an iterable to all the edges in the graph that have a particular key/value property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Overridable Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

the key of the edge

value

Type: [System.Object](#)

the value of the edge

Return Value

Type: [IEnumerable\(IEdge\)](#)

an iterable of edges with provided key and value

VelocityDB Class Library

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[Graph Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

Graph.GetEdges(*T*) Method (String, *T*)

Return an iterable to all the edges in the graph that have a particular key/value property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IEdge> GetEdges<T>(
    string key,
    T value
)
where T : IComparable
```

VB

```
Public Overridable Function GetEdges(Of T As IComparable) (
    key As String,
    value As T
) As IEnumerable(Of IEdge)
```

C++

```
public:
    generic<typename T>
    where T : IComparable
    virtual IEnumerable<IEdge^>^ GetEdges (
        String^ key,
        T value
    )
```

F#

```
abstract GetEdges :
    key : string *
    value : 'T -> IEnumerable<IEdge> when 'T : IComparable
override GetEdges :
    key : string *
    value : 'T -> IEnumerable<IEdge> when 'T : IComparable
```

Parameters

key

Type: [System.String](#)

the key of the edge

value

Type: *T*

the value of the edge

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Graph.GetEdges`1(System.String,`0)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

an iterable of edges with provided key and value

See Also

[Graph Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

Graph.EdgeType Method

Get a [EdgeType](#) associated with an [Edge](#) id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType GetEdgeType (  
    int id  
)
```

VB

```
Public Function GetEdgeType (  
    id As Integer  
) As EdgeType
```

C++

```
public:  
EdgeType^ GetEdgeType (  
    int id  
)
```

F#

```
member GetEdgeType :  
    id : int -> EdgeType
```

Parameters

id

Type: [System.Int32](#)

Id to look for

Return Value

Type: [EdgeType](#)

the associated edge type

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetValues Method

Enumerates all the values for the given vertex/edge property

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IComparable> GetValues (  
    PropertyType property  
)
```

VB

```
Public Function GetValues (  
    property As PropertyType  
) As IEnumerable(Of IComparable)
```

C++

```
public:  
IEnumerable<IComparable^> GetValues (  
    PropertyType^ property  
)
```

F#

```
member GetValues :  
    property : PropertyType -> IEnumerable<IComparable>
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

Edge or Vertex property

Return Value

Type: [IEnumerable\(IComparable\)](#)

Enumeration of property values



See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetVertex Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|  | GetVertex(Int32) | Look up a Vertex |
|  | GetVertex(Object) | Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetVertex Method (Int32)

Look up a [Vertex](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex GetVertex(  
    int id  
)
```

VB

```
Public Function GetVertex (  
    id As Integer  
) As Vertex
```

C++

```
public:  
Vertex^ GetVertex(  
    int id  
)
```

F#

```
member GetVertex :  
    id : int -> Vertex
```

Parameters

id

Type: [System.Int32](#)

Id of [Vertex](#) to lookup

Return Value

Type: [Vertex](#)

A vertex with the matching id

See Also

[Graph Class](#)

[GetVertex Overload](#)

[VelocityGraph Namespace](#)

Graph.GetVertex Method (Object)

Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

the identifier of the vertex to retrieved from the graph, must be a UInt64

Return Value

Type: [IVertex](#)

the vertex referenced by the provided identifier or null when no such vertex exists

Implements

[IGraph.GetVertex\(Object\)](#)

See Also

[Graph Class](#)

[GetVertex Overload](#)

VelocityDB Class Library

[VelocityGraph Namespace](#)

Graph.GetVertexType Method

Given a [Vertex](#) id, returns the corresponding [Vertex](#).

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType GetVertexType(  
    int id  
)
```

VB

```
Public Function GetVertexType (  
    id As Integer  
) As VertexType
```

C++

```
public:  
VertexType^ GetVertexType(  
    int id  
)
```

F#

```
member GetVertexType :  
    id : int -> VertexType
```

Parameters

id

Type: [System.Int32](#)

Id of a [Vertex](#)

Return Value

Type: [VertexType](#)

A [VertexType](#) with id. Throws [VertexTypeDoesNotExistException](#) if [Vertex](#) does not exist



See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|------------------------------------------------------------------------------------------------|
|  | GetVertices() | Return an iterable to all the vertices in the graph. |
|  | GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.GetVertices Method

Return an iterable to all the vertices in the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

an iterable reference to all vertices in the graph

Implements

[IGraph.GetVertices\(\)](#)

See Also

[Graph Class](#)

[GetVertices Overload](#)

[VelocityGraph Namespace](#)

Graph.GetVertices Method (String, Object)

Return an iterable to all the vertices in the graph that have a particular key/value property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetVertices (  
    key As String,  
    value As Object  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>  
override GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

the key of vertex

value

Type: [System.Object](#)

the value of the vertex

Return Value

Type: [IEnumerable<IVertex>](#)

an iterable of vertices with provided key and value

VelocityDB Class Library

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also



[Graph Class](#)

[GetVertices Overload](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphJson Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------|--------------------------------------------------------------------------------------------------------|
|  | ImportGraphJson(String) | Input the GraphJson JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | ImportGraphJson(Stream, Int32) | Input the GrapJson JSON stream data into the graph. In practice, usually the provided graph is empty. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphJson Method (String)

Input the GraphJson JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ImportGraphJson(  
    string filename  
)
```

VB

```
Public Sub ImportGraphJson (  
    filename As String  
)
```

C++

```
public:  
void ImportGraphJson(  
    String^ filename  
)
```

F#

```
member ImportGraphJson :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file of JSON data

See Also

[Graph Class](#)

[ImportGraphJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphJson Method (Stream, Int32)

Input the GrapJson JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ImportGraphJson(  
    Stream grapJsonInputStream,  
    int bufferSize = 1000  
)
```

VB

```
Public Sub ImportGraphJson (  
    grapJsonInputStream As Stream,  
    Optional bufferSize As Integer = 1000  
)
```

C++

```
public:  
void ImportGraphJson(  
    Stream^ grapJsonInputStream,  
    int bufferSize = 1000  
)
```

F#

```
member ImportGraphJson :  
    grapJsonInputStream : Stream *  
    ?bufferSize : int  
(* Defaults:  
    let _bufferSize = defaultArg bufferSize 1000  
)  
-> unit
```

Parameters

grapJsonInputStream

Type: [System.IO.Stream](#)

an Stream of GraphJson JSON data

bufferSize (Optional)

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions

See Also

[Graph Class](#)



VelocityDB Class Library

[ImportGraphJson Overload](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphSon Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  | ImportGraphSon(String) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | ImportGraphSon(Stream, Int32) | Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphSon Method (String)

Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ImportGraphSon(  
    string filename  
)
```

VB

```
Public Sub ImportGraphSon (  
    filename As String  
)
```

C++

```
public:  
void ImportGraphSon(  
    String^ filename  
)
```

F#

```
member ImportGraphSon :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file of JSON data

See Also

[Graph Class](#)

[ImportGraphSon Overload](#)

[VelocityGraph Namespace](#)

Graph.ImportGraphSon Method (Stream, Int32)

Input the GraphSon JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ImportGraphSon(  
    Stream jsonInputStream,  
    int bufferSize  
)
```

VB

```
Public Sub ImportGraphSon (  
    jsonInputStream As Stream,  
    bufferSize As Integer  
)
```

C++

```
public:  
void ImportGraphSon(  
    Stream^ jsonInputStream,  
    int bufferSize  
)
```

F#

```
member ImportGraphSon :  
    jsonInputStream : Stream *  
    bufferSize : int -> unit
```

Parameters

jsonInputStream

Type: [System.IO.Stream](#)

an Stream of JSON data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions

See Also



[Graph Class](#)

[ImportGraphSon Overload](#)

[VelocityGraph Namespace](#)

Graph.NewEdge Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | NewEdge(EdgeType, Vertex, Vertex) | Creates a new edge instance. |
|  | NewEdge(EdgeType, PropertyType, Object, PropertyType, Object) | Creates a new edge instance. The tail of the edge will be any node having the given tailV Value for the given tailAttr Property identifier, and the head of the edge will be any node having the given headV Value for the given headAttr Property identifier. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.NewEdge Method (EdgeType, Vertex, Vertex)

Creates a new edge instance.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge NewEdge(  
    EdgeType edgeType,  
    Vertex tail,  
    Vertex head  
)
```

VB

```
Public Function NewEdge (  
    edgeType As EdgeType,  
    tail As Vertex,  
    head As Vertex  
) As Edge
```

C++

```
public:  
Edge^ NewEdge (  
    EdgeType^ edgeType,  
    Vertex^ tail,  
    Vertex^ head  
)
```

F#

```
member NewEdge :  
    edgeType : EdgeType *  
    tail : Vertex *  
    head : Vertex -> Edge
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

Edge type identifier.

tail

Type: [VelocityGraph.Vertex](#)

Source OID.

head

Type: [VelocityGraph.Vertex](#)

Target OID.

VelocityDB Class Library

Return Value

Type: [Edge](#)

Unique OID of the new edge instance.

See Also

[Graph Class](#)

[NewEdge Overload](#)

[VelocityGraph Namespace](#)

Graph.NewEdge Method (EdgeType, PropertyType, Object, PropertyType, Object)

Creates a new edge instance. The tail of the edge will be any node having the given tailV Value for the given tailAttr Property identifier, and the head of the edge will be any node having the given headV Value for the given headAttr Property identifier.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge NewEdge (
    EdgeType edgeType,
    PropertyType tailAttr,
    Object tailV,
    PropertyType headAttr,
    Object headV
)
```

VB

```
Public Function NewEdge (
    edgeType As EdgeType,
    tailAttr As PropertyType,
    tailV As Object,
    headAttr As PropertyType,
    headV As Object
) As IEdge
```

C++

```
public:
IEdge^ NewEdge (
    EdgeType^ edgeType,
    PropertyType^ tailAttr,
    Object^ tailV,
    PropertyType^ headAttr,
    Object^ headV
)
```

F#

```
member NewEdge :
    edgeType : EdgeType *
    tailAttr : PropertyType *
    tailV : Object *
    headAttr : PropertyType *
    headV : Object -> IEdge
```

VelocityDB Class Library

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

Node or edge type identifier.

tailAttr

Type: [VelocityGraph.PropertyType](#)

Property identifier.

tailV

Type: [System.Object](#)

Tail value

headAttr

Type: [VelocityGraph.PropertyType](#)

Property identifier.

headV

Type: [System.Object](#)

Head value

Return Value

Type: [IEdge](#)

Unique edge instance.

See Also

[Graph Class](#)

[NewEdge Overload](#)

[VelocityGraph Namespace](#)

Graph.NewEdgeProperty Method

Creates a new Property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType NewEdgeProperty(  
    EdgeType edgeType,  
    string name,  
    DataType dt,  
    PropertyKind kind  
)
```

VB

```
Public Function NewEdgeProperty (  
    edgeType As EdgeType,  
    name As String,  
    dt As DataType,  
    kind As PropertyKind  
) As PropertyType
```

C++

```
public:  
PropertyType^ NewEdgeProperty(  
    EdgeType^ edgeType,  
    String^ name,  
    DataType dt,  
    PropertyKind kind  
)
```

F#

```
member NewEdgeProperty :  
    edgeType : EdgeType *  
    name : string *  
    dt : DataType *  
    kind : PropertyKind -> PropertyType
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

Node or edge type identifier.

name

Type: [System.String](#)

Unique name for the new Property.

VelocityDB Class Library

dt

Type: [VelocityGraph.DataType](#)

Data type for the new Property.

kind

Type: [VelocityGraph.PropertyKind](#)

Property kind.

Return Value

Type: [PropertyType](#)

Unique Property identifier.


See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.NewEdgeType Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------|--------------------------|
|  | NewEdgeType(String, Boolean, EdgeType) | Creates a new edge type. |
|  | NewEdgeType(String, Boolean, VertexType, VertexType, EdgeType) | Creates a new edge type. |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.NewEdgeType Method (String, Boolean, EdgeType)

Creates a new edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType NewEdgeType(  
    string name,  
    bool bidirectional,  
    EdgeType baseType = null  
)
```

VB

```
Public Function NewEdgeType (  
    name As String,  
    bidirectional As Boolean,  
    Optional baseType As EdgeType = Nothing  
) As EdgeType
```

C++

```
public:  
EdgeType^ NewEdgeType(  
    String^ name,  
    bool bidirectional,  
    EdgeType^ baseType = nullptr  
)
```

F#

```
member NewEdgeType :  
    name : string *  
    bidirectional : bool *  
    ?baseType : EdgeType  
(* Defaults:  
    let_baseType = defaultArg baseType null  
*)  
-> EdgeType
```

Parameters

name

Type: [System.String](#)

Unique name for the new edge type.

bidirectional

Type: [System.Boolean](#)

If true, this creates a bidirectional edge type, otherwise this creates a unidirectional edge type.

VelocityDB Class Library

baseType (Optional)

Type: [VelocityGraph.EdgeType](#)

Base EdgeType for the new EdgeType.

Return Value

Type: [EdgeType](#)

Unique edge type.

Return Value

Type: [EdgeType](#)

a new edge type

See Also

[Graph Class](#)

[NewEdgeType Overload](#)

[VelocityGraph Namespace](#)

Graph.NewEdgeType Method (String, Boolean, VertexType, VertexType, EdgeType)

Creates a new edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeType NewEdgeType (  
    string name,  
    bool bidirectional,  
    VertexType tailType,  
    VertexType headType,  
    EdgeType baseType = null  
)
```

VB

```
Public Function NewEdgeType (  
    name As String,  
    bidirectional As Boolean,  
    tailType As VertexType,  
    headType As VertexType,  
    Optional baseType As EdgeType = Nothing  
) As EdgeType
```

C++

```
public:  
EdgeType^ NewEdgeType (  
    String^ name,  
    bool bidirectional,  
    VertexType^ tailType,  
    VertexType^ headType,  
    EdgeType^ baseType = nullptr  
)
```

F#

```
member NewEdgeType :  
    name : string *  
    bidirectional : bool *  
    tailType : VertexType *  
    headType : VertexType *  
    ?baseType : EdgeType  
(* Defaults:  
    let_baseType = defaultArg baseType null  
*)  
-> EdgeType
```

VelocityDB Class Library

Parameters

name

Type: [System.String](#)

Unique name for the new edge type.

bidirectional

Type: [System.Boolean](#)

If true, this creates a bidirectional edge type, otherwise this creates a unidirectional edge type.

tailType

Type: [VelocityGraph.VertexType](#)

a fixed tail VertexType

headType

Type: [VelocityGraph.VertexType](#)

a fixed head VertexType

baseType (Optional)

Type: [VelocityGraph.EdgeType](#)

if specified make the new [EdgeType](#) a subclass of this [EdgeType](#)

Return Value

Type: [EdgeType](#)

Unique edge type.

Return Value

Type: [EdgeType](#)

a new edge type

See Also

[Graph Class](#)

[NewEdgeType Overload](#)

[VelocityGraph Namespace](#)

Graph.NewVertex Method

Creates a new node instance.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex NewVertex(  
    VertexType vertexType,  
    int vId = 0  
)
```

VB

```
Public Function NewVertex (  
    vertexType As VertexType,  
    Optional vId As Integer = 0  
) As Vertex
```

C++

```
public:  
Vertex^ NewVertex(  
    VertexType^ vertexType,  
    int vId = 0  
)
```

F#

```
member NewVertex :  
    vertexType : VertexType *  
    ?vId : int  
(* Defaults:  
    let _vId = defaultArg vId 0  
)  
-> Vertex
```

Parameters

vertexType

Type: [VelocityGraph.VertexType](#)

Node type identifier.

vId (Optional)

Type: [System.Int32](#)

[Missing <param name="vId"/> documentation for "M:VelocityGraph.Graph.NewVertex(VelocityGraph.VertexType,System.Int32)"]

VelocityDB Class Library

Return Value

Type: [Vertex](#)

Unique OID of the new node instance.

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.NewVertexProperty Method

Creates a new Property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType NewVertexProperty(  
    VertexType vertexType,  
    string name,  
    DataType dt,  
    PropertyKind kind  
)
```

VB

```
Public Function NewVertexProperty (  
    vertexType As VertexType,  
    name As String,  
    dt As DataType,  
    kind As PropertyKind  
) As PropertyType
```

C++

```
public:  
PropertyType^ NewVertexProperty(  
    VertexType^ vertexType,  
    String^ name,  
    DataType dt,  
    PropertyKind kind  
)
```

F#

```
member NewVertexProperty :  
    vertexType : VertexType *  
    name : string *  
    dt : DataType *  
    kind : PropertyKind -> PropertyType
```

Parameters

vertexType

Type: [VelocityGraph.VertexType](#)

Node or edge type identifier.

name

Type: [System.String](#)

Unique name for the new Property.

VelocityDB Class Library

dt

Type: [VelocityGraph.DataType](#)

Data type for the new Property.

kind

Type: [VelocityGraph.PropertyKind](#)

Property kind.

Return Value

Type: [PropertyType](#)

Unique Property identifier.

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.NewVertexType Method

Creates a new node type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType NewVertexType(  
    string name,  
    VertexType baseType = null  
)
```

VB

```
Public Function NewVertexType (  
    name As String,  
    Optional baseType As VertexType = Nothing  
) As VertexType
```

C++

```
public:  
VertexType^ NewVertexType (  
    String^ name,  
    VertexType^ baseType = nullptr  
)
```

F#

```
member NewVertexType :  
    name : string *  
    ?baseType : VertexType  
(* Defaults:  
    let _baseType = defaultArg baseType null  
*)  
-> VertexType
```

Parameters

name

Type: [System.String](#)

Unique name for the new vertex type.

baseType (Optional)

Type: [VelocityGraph.VertexType](#)

Base VertexType for the new VertexType.

Return Value

Type: [VertexType](#)

Unique graph type identifier.

VelocityDB Class Library

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Open Method

Opens a graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Graph Open(  
    SessionBase session,  
    int graphInstance = 0  
)
```

VB

```
Public Shared Function Open (  
    session As SessionBase,  
    Optional graphInstance As Integer = 0  
) As Graph
```

C++

```
public:  
static Graph^ Open(  
    SessionBase^ session,  
    int graphInstance = 0  
)
```

F#

```
static member Open :  
    session : SessionBase *  
    ?graphInstance : int  
(* Defaults:  
    let_graphInstance = defaultArg graphInstance 0  
)  
-> Graph
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

session to be associated with the opened graph

graphInstance (Optional)

Type: [System.Int32](#)

if multiple graphs exist, choose which one to open

Return Value

Type: [Graph](#)

A graph

VelocityDB Class Library

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Persist Method

Persists this object. Override in your subclasses when you want fields of your class to be persisted in some special way.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override ulong Persist(
    Placement place,
    SessionBase session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable> toPersist = null
)
```

VB

```
Public Overrides Function Persist (
    place As Placement,
    session As SessionBase,
    Optional persistRefs As Boolean = true,
    Optional disableFlush As Boolean = false,
    Optional toPersist As Queue(Of IOptimizedPersistable) = Nothing
) As ULong
```

C++

```
public:
virtual unsigned long long Persist(
    Placement^ place,
    SessionBase^ session,
    bool persistRefs = true,
    bool disableFlush = false,
    Queue<IOptimizedPersistable^>^ toPersist = nullptr
) override
```

F#

```
abstract Persist :
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let _persistRefs = defaultArg persistRefs true
    let _disableFlush = defaultArg disableFlush false
    let _toPersist = defaultArg toPersist null
*)
-> uint64
override Persist :
```

```
    place : Placement *
    session : SessionBase *
    ?persistRefs : bool *
    ?disableFlush : bool *
    ?toPersist : Queue<IOptimizedPersistable>
(* Defaults:
    let_persistRefs = defaultArg persistRefs true
    let_disableFlush = defaultArg disableFlush false
    let_toPersist = defaultArg toPersist null
*)
-> uint64
```

Parameters

place

Type: [VelocityDb.Placement](#)

The placement rules to follow when persisting this object

session

Type: [VelocityDb.Session.SessionBase](#)

The session managing this object

persistRefs (Optional)

Type: [System.Boolean](#)

If true, objects referenced from this object will also be persisted

disableFlush (Optional)

Type: [System.Boolean](#)

If true, disables possible flushing of updated pages while persisting this object; otherwise page flushing may occur

toPersist (Optional)

Type: [System.Collections.Generic.Queue<IOptimizedPersistable>](#)

A queue of objects remaining to be persisted. Pass as a parameter to session.Persist

Return Value

Type: [UInt64](#)

The object id of this persistent object

Implements

[IOptimizedPersistable.Persist\(Placement, SessionBase, Boolean, Boolean, Queue<IOptimizedPersistable>\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Query Method

Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

a graph query object with methods for constraining which data is pulled from the underlying graph

Implements

[IGraph.Query\(\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.RemoveEdge Method

Remove the provided edge from the graph.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (
    IEdge edge
)
```

VB

```
Public Sub RemoveEdge (
    edge As IEdge
)
```

C++

```
public:
virtual void RemoveEdge (
    IEdge^ edge
) sealed
```

F#

```
abstract RemoveEdge :
    edge : IEdge -> unit
override RemoveEdge :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the edge to remove from the graph

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.RemovePropertyType Method

Removes a property type except if any vertex or edge is using it

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemovePropertyType (
    PropertyType pt
)
```

VB

```
Public Sub RemovePropertyType (
    pt As PropertyType
)
```

C++

```
public:
void RemovePropertyType (
    PropertyType^ pt
)
```

F#

```
member RemovePropertyType :
    pt : PropertyType -> unit
```

Parameters

pt

Type: [VelocityGraph.PropertyType](#)

property type to remove

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.RemoveVertex Method

Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex to remove from the graph

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.RemoveVertexType Method

Removes a [VertexType](#) from this graph. An exception is thrown if the [VertexType](#) is in use.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertexType (  
    VertexType type  
)
```

VB

```
Public Sub RemoveVertexType (  
    type As VertexType  
)
```

C++

```
public:  
void RemoveVertexType (  
    VertexType^ type  
)
```

F#

```
member RemoveVertexType :  
    type : VertexType -> unit
```

Parameters

type

Type: [VelocityGraph.VertexType](#)

a [VertexType](#) instance



See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Select Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | Select(PropertyType, Func(IComparable, IComparable, Boolean), IComparable) | Enumerates all elements satisfying the given condition for the given property and value |
|  | Select(PropertyType, Func(IComparable, IComparable, IComparable, Boolean), IComparable, IComparable) | numerates all elements satisfying the given condition for the given property and value range |

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Select Method (PropertyType, Func(IComparable, IComparable, Boolean), IComparable)

Enumerates all elements satisfying the given condition for the given property and value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Select(  
    PropertyType property,  
    Func<IComparable, IComparable, bool> condition,  
    IComparable value  
)
```

VB

```
Public Function Select (  
    property As PropertyType,  
    condition As Func(Of IComparable, IComparable, Boolean),  
    value As IComparable  
) As IEnumerable(Of IElement)
```

C++

```
public:  
IEnumerable<IElement>^ Select(  
    PropertyType^ property,  
    Func<IComparable^, IComparable^, bool>^ condition,  
    IComparable^ value  
)
```

F#

```
member Select :  
    property : PropertyType *  
    condition : Func<IComparable, IComparable, bool> *  
    value : IComparable -> IEnumerable<IElement>
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

Property we are looking for

condition

Type: [System.Func\(IComparable, IComparable, Boolean\)](#)

A filter function, applied with in graph value as 1st parameter and value as second parameter.

value

VelocityDB Class Library

Type: [System.IComparable](#)

Filtering value

Return Value

Type: [IEnumerable\(IElement\)](#)

Enum of IElement

See Also

[Graph Class](#)

[Select Overload](#)

[VelocityGraph Namespace](#)

Graph.Select Method (PropertyType, Func(IComparable, IComparable, IComparable, Boolean), IComparable, IComparable)

numerates all elements satisfying the given condition for the given property and value range

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Select(  
    PropertyType property,  
    Func<IComparable, IComparable, IComparable, bool> condition,  
    IComparable lower,  
    IComparable higher  
)
```

VB

```
Public Function Select (  
    property As PropertyType,  
    condition As Func(Of IComparable, IComparable, IComparable, Boolean),  
    lower As IComparable,  
    higher As IComparable  
) As IEnumerable(Of IElement)
```

C++

```
public:  
IEnumerable<IElement>^ Select(  
    PropertyType^ property,  
    Func<IComparable^, IComparable^, IComparable^, bool>^ condition,  
    IComparable^ lower,  
    IComparable^ higher  
)
```

F#

```
member Select :  
    property : PropertyType *  
    condition : Func<IComparable, IComparable, IComparable, bool> *  
    lower : IComparable *  
    higher : IComparable -> IEnumerable<IElement>
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

Property we are looking for

condition

Type: [System.Func\(IComparable, IComparable, IComparable, Boolean\)](#)

filter function, applied with in graph value as 1st parameter, lower as 2nd and higher as 3rd parameter.

lower

Type: [System.IComparable](#)

lower value in filtering

higher

Type: [System.IComparable](#)

higher value in filtering

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Graph.Select(VelocityGraph.PropertyType,System.Func{System.IComparable,System.IComparable,System.IComparable,System.Boolean},System.IComparable,System.IComparable)"]

See Also

[Graph Class](#)

[Select Overload](#)

[VelocityGraph Namespace](#)

Graph.Shutdown Method

A shutdown function is required to properly close the graph. This is important for implementations that utilize disk-based serializations.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.ToString Method

Displays class name plus object id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Traverse Method

Selects all neighbor Vertices from or to each of the node OID in the given collection and for the given edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Dictionary<Vertex, HashSet<Edge>> Traverse (
    Vertex[] vertices,
    EdgeType etype,
    Direction dir
)
```

VB

```
Public Function Traverse (
    vertices As Vertex(),
    etype As EdgeType,
    dir As Direction
) As Dictionary(Of Vertex, HashSet(Of Edge))
```

C++

```
public:
    Dictionary<Vertex^, HashSet<Edge^>> Traverse (
        array<Vertex^> vertices,
        EdgeType^ etype,
        Direction dir
    )
```

F#

```
member Traverse :
    vertices : Vertex[] *
    etype : EdgeType *
    dir : Direction -> Dictionary<Vertex, HashSet<Edge>>
```

Parameters

vertices

Type: [VelocityGraph.Vertex\[\]](#)

Vertex collection.

etype

Type: [VelocityGraph.EdgeType](#)

Edge type identifier.

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Direction.

Return Value

Type: [Dictionary\(Vertex, HashSet\(Edge\)\)](#)

Dictionary of vertex keys with edges path to vertex

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

Graph.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also

[Graph Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings Class

[Missing <summary> documentation for "T:VelocityGraph.GraphJsonSettings"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.GraphJsonSettings

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphJsonSettings
```

VB

```
Public Class GraphJsonSettings
```

C++


```
public ref class GraphJsonSettings
```

F#









```
type GraphJsonSettings = class end
```

The **GraphJsonSettings** type exposes the following members.




Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | GraphJsonSettings | Initializes a new instance of the GraphJsonSettings class |

Properties

| | Name | Description |
|----------------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | ClusterFuncProp | |
|  | ClusterProp | |
|  S | Default | |
|  | EdgeCaptionProp | |
|  | EdgeTypeFuncProp | |
|  | IdProp | |
|  | NodeCaptionProp | |
|  | SourceProp | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------------|--|
|  | TargetProp | |
|  | TypeProp | |
|  | VertexTypeFuncProp | |

See Also

[VelocityGraph Namespace](#)

GraphJsonSettings Constructor

Initializes a new instance of the [GraphJsonSettings](#) class

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphJsonSettings ()
```

VB

```
Public Sub New
```

C++

```
public:  
GraphJsonSettings ()
```

F#

```
new : unit -> GraphJsonSettings
```

See Also












[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.GraphJsonSettings Properties

The [GraphJsonSettings](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------|
|  | ClusterFuncProp | |
|  | ClusterProp | |
|  | Default | |
|  | EdgeCaptionProp | |
|  | EdgeTypeFuncProp | |
|  | IdProp | |
|  | NodeCaptionProp | |
|  | SourceProp | |
|  | TargetProp | |
|  | TypeProp | |
|  | VertexTypeFuncProp | |

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.ClusterFuncProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.ClusterFuncProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Func<IVertex, int> ClusterFuncProp { get; set; }
```

VB

```
Public Property ClusterFuncProp As Func(Of IVertex, Integer)  
    Get  
    Set
```

C++

```
public:  
property Func<IVertex^, int>^ ClusterFuncProp {  
    Func<IVertex^, int>^ get ();  
    void set (Func<IVertex^, int>^ value);  
}
```

F#

```
member ClusterFuncProp : Func<IVertex, int> with get, set
```

Property Value

Type: [Func\(IVertex, Int32\)](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.ClusterProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.ClusterProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string ClusterProp { get; set; }
```

VB

```
Public Property ClusterProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ ClusterProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member ClusterProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.Default Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.Default"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static GraphJsonSettings Default { get; }
```

VB

```
Public Shared ReadOnly Property Default As GraphJsonSettings  
    Get
```

C++

```
public:  
static property GraphJsonSettings^ Default {  
    GraphJsonSettings^ get ();  
}
```

F#

```
static member Default : GraphJsonSettings with get
```

Property Value

Type: [GraphJsonSettings](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.EdgeCaptionProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.EdgeCaptionProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeCaptionProp { get; set; }
```

VB

```
Public Property EdgeCaptionProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeCaptionProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeCaptionProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.EdgeTypeFuncProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.EdgeTypeFuncProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Func<IEdge, string> EdgeTypeFuncProp { get; set; }
```

VB

```
Public Property EdgeTypeFuncProp As Func(Of IEdge, String)  
    Get  
    Set
```

C++

```
public:  
property Func<IEdge^, String^^> EdgeTypeFuncProp {  
    Func<IEdge^, String^^> get ();  
    void set (Func<IEdge^, String^^> value);  
}
```

F#

```
member EdgeTypeFuncProp : Func<IEdge, string> with get, set
```

Property Value

Type: [Func\(IEdge, String\)](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.IdProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.IdProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string IdProp { get; set; }
```

VB

```
Public Property IdProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ IdProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member IdProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.NodeCaptionProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.NodeCaptionProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string NodeCaptionProp { get; set; }
```

VB

```
Public Property NodeCaptionProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ NodeCaptionProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member NodeCaptionProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.SourceProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.SourceProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SourceProp { get; set; }
```

VB

```
Public Property SourceProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ SourceProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member SourceProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.TargetProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.TargetProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TargetProp { get; set; }
```

VB

```
Public Property TargetProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ TargetProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member TargetProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.TypeProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.TypeProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TypeProp { get; set; }
```

VB

```
Public Property TypeProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ TypeProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member TypeProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

GraphJsonSettings.VertexTypeFuncProp Property

[Missing <summary> documentation for "P:VelocityGraph.GraphJsonSettings.VertexTypeFuncProp"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Func<IVertex, string> VertexTypeFuncProp { get; set; }
```

VB

```
Public Property VertexTypeFuncProp As Func(Of IVertex, String)  
    Get  
    Set
```

C++

```
public:  
property Func<IVertex^, String^>^ VertexTypeFuncProp {  
    Func<IVertex^, String^>^ get ();  
    void set (Func<IVertex^, String^>^ value);  
}
```

F#

```
member VertexTypeFuncProp : Func<IVertex, string> with get, set
```

Property Value

Type: [Func\(IVertex, String\)](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph Namespace](#)

PropertyKind Enumeration

A property can be index or not or index can require each entry to be unique

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum PropertyKind
```

VB

```
Public Enumeration PropertyKind
```

C++

```
public enum class PropertyKind
```

F#

```
type PropertyKind
```

Members

| Member name | Value | Description |
|-------------------|-------|---------------------------------------------|
| Indexed | 0 | Add to index |
| Unique | 1 | Add to index and require value to be unique |
| NotIndexed | 2 | Don't index |

See Also

[VelocityGraph Namespace](#)

PropertyType Class

Keeper of [Edge](#) and [Vertex](#) properties

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityGraph.PropertyType

[VelocityGraph.PropertyTypeT\(T\)](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public abstract class PropertyType : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public MustInherit Class PropertyType
    Inherits OptimizedPersistable
```

C++






```
[SerializableAttribute]
public ref class PropertyType abstract : public OptimizedPersistable
```

F#

```
[<AbstractClassAttribute>]
[<SerializableAttribute>]
type PropertyType =
    class
        inherit OptimizedPersistable
    end
```











The **PropertyType** type exposes the following members.

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------|
|  | IsVertexProperty | Is this a Vertex property, if not then it is an Edge property |
|  | MyGraph | |
|  | Name | Get name of this property |
|  | PropertyId | Get id of this property type |
|  | TypeId | Id of a Vertex or Edge |

| | |
|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
|  ValueType | Get the type of property values for this property |
|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------|

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------|
|  | GetPropertyEdge | Try to find an Edge with a given property value. |
|  | GetPropertyEdges | Try to find all Edge with a given property value. |
|  | GetPropertyValue | Get property value of a Vertex/Edge |
|  | GetPropertyVertex | |
|  | GetPropertyVertices(Boolean) | |
|  | GetPropertyVertices(Comparable, Boolean) | Try to find all Vertex with a given property value. |
|  | HasPropertyValue | Check if an element has a property value |
|  | RemovePropertyValue | Remove a property value |
|  | SetPropertyValue(EdgeType, Int32, Int32, Comparable) | |
|  | SetPropertyValue(VertexType, Int32, Int32, Comparable) | Sets a property value for an element |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |







See Also

[VelocityGraph Namespace](#)

PropertyType.PropertyType Properties

The [PropertyType](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------------------------------------|
|  | IsVertexProperty | Is this a Vertex property, if not then it is an Edge property |
|  | MyGraph | |
|  | Name | Get name of this property |
|  | PropertyId | Get id of this property type |
|  | TypeId | Id of a Vertex or Edge |
|  | ValueType | Get the type of property values for this property |

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.IsVertexProperty Property

Is this a [Vertex](#) property, if not then it is an [Edge](#) property

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsVertexProperty { get; }
```

VB

```
Public ReadOnly Property IsVertexProperty As Boolean  
    Get
```

C++

```
public:  
property bool IsVertexProperty {  
    bool get ();  
}
```

F#

```
member IsVertexProperty : bool with get
```

Property Value

Type: [Boolean](#)

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.MyGraph Property

[Missing <summary> documentation for "P:VelocityGraph.PropertyType.MyGraph"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Graph MyGraph { get; }
```

VB

```
Public ReadOnly Property MyGraph As Graph  
    Get
```

C++

```
public:  
property Graph^ MyGraph {  
    Graph^ get ();  
}
```

F#

```
member MyGraph : Graph with get
```

Property Value

Type: [Graph](#)

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.Name Property

Get name of this property

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
property String^ Name {  
    String^ get ();  
}
```

F#

```
member Name : string with get
```

Property Value

Type: [String](#)

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.PropertyId Property

Get id of this property type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int PropertyId { get; }
```

VB

```
Public ReadOnly Property PropertyId As Integer  
    Get
```

C++

```
public:  
property int PropertyId {  
    int get ();  
}
```

F#

```
member PropertyId : int with get
```

Property Value

Type: [Int32](#)

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.TypeId Property

Id of a [Vertex](#) or [Edge](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int TypeId { get; }
```

VB

```
Public ReadOnly Property TypeId As Integer  
    Get
```

C++

```
public:  
property int TypeId {  
    int get ();  
}
```

F#

```
member TypeId : int with get
```

Property Value

Type: [Int32](#)

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.ValueType Property

Get the type of property values for this property

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Type ValueType { get; }
```

VB

```
Public MustOverride ReadOnly Property ValueType As Type  
    Get
```

C++

```
public:  
virtual property Type^ ValueType {  
    Type^ get () abstract;  
}
```

F#

```
abstract ValueType : Type with get
```

Property Value

Type: [Type](#)

See Also











[PropertyType Class](#)

[VelocityGraph Namespace](#)



PropertyType.PropertyType Methods

The [PropertyType](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|
|  GetPropertyEdge | Try to find an Edge with a given property value. |
|  GetPropertyEdges | Try to find all Edge with a given property value. |
|  GetPropertyValue | Get property value of a Vertex/Edge |
|  GetPropertyVertex | |
|  GetPropertyVertices(Boolean) | |
|  GetPropertyVertices(Comparable, Boolean) | Try to find all Vertex with a given property value. |
|  HasPropertyValue | Check if an element has a property value |
|  RemovePropertyValue | Remove a property value |
|  SetPropertyValue(EdgeType, Int32, Int32, Comparable) | |
|  SetPropertyValue(VertexType, Int32, Int32, Comparable) | Sets a property value for an element |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyEdge Method

Try to find an [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Edge GetPropertyEdge (
    IComparable value
)
```

VB

```
Public MustOverride Function GetPropertyEdge (
    value As IComparable
) As Edge
```

C++

```
public:
virtual Edge^ GetPropertyEdge (
    IComparable^ value
) abstract
```

F#

```
abstract GetPropertyEdge :
    value : IComparable -> Edge
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [Edge](#)

An edge with a matching property value

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyEdges Method

Try to find all [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<Edge> GetPropertyEdges (
    IComparable value
)
```

VB

```
Public MustOverride Function GetPropertyEdges (
    value As IComparable
) As IEnumerable(Of Edge)
```

C++

```
public:
virtual IEnumerable<Edge^>^ GetPropertyEdges (
    IComparable^ value
) abstract
```

F#

```
abstract GetPropertyEdges :
    value : IComparable -> IEnumerable<Edge>
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [IEnumerable\(Edge\)](#)

An edge with a matching property value

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyValue Method

Get property value of a Vertex/Edge

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IComparable GetPropertyValue(  
    int elementId  
)
```

VB

```
Public MustOverride Function GetPropertyValue (  
    elementId As Integer  
) As IComparable
```

C++

```
public:  
virtual IComparable^ GetPropertyValue(  
    int elementId  
) abstract
```

F#

```
abstract GetPropertyValue :  
    elementId : int -> IComparable
```

Parameters

elementId

Type: [System.Int32](#)

Id of a Vertex/Edge

Return Value

Type: [IComparable](#)

the property value

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.PropertyType.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Vertex GetPropertyVertex(  
    IComparable value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public MustOverride Function GetPropertyVertex (  
    value As IComparable,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
virtual Vertex^ GetPropertyVertex(  
    IComparable^ value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
) abstract
```

F#

```
abstract GetPropertyVertex :  
    value : IComparable *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
    let_errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex
```

Parameters

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

polymorphic (Optional)

Type: [System.Boolean](#)

[Missing <param name="polymorphic"/> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

errorIfNotFound (Optional)

Type: [System.Boolean](#)

[Missing <param name="errorIfNotFound"/> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

Return Value

Type: [Vertex](#)

[Missing <returns> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]



See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------|---------------------------------------------------------------------|
|  | GetPropertyVertices(Boolean) | |
|  | GetPropertyVertices(Comparable, Boolean) | Try to find all Vertex with a given property value. |

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyVertices Method (Boolean)

[Missing <summary> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertices(System.Boolean)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<Vertex> GetPropertyVertices (
    bool polymorphic = false
)
```

VB

```
Public MustOverride Function GetPropertyVertices (
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
virtual IEnumerable<Vertex^>^ GetPropertyVertices (
    bool polymorphic = false
) abstract
```

F#

```
abstract GetPropertyVertices :
    ?polymorphic : bool
(* Defaults:
    let polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

polymorphic (Optional)

Type: [System.Boolean](#)

[Missing <param name="polymorphic"/> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertices(System.Boolean)"]

Return Value

Type: [IEnumerable\(Vertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.PropertyType.GetPropertyVertices(System.Boolean)"]

See Also

[PropertyType Class](#)

VelocityDB Class Library

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyType.GetPropertyVertices Method (IComparable, Boolean)

Try to find all [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<Vertex> GetPropertyVertices (
    IComparable value,
    bool polymorphic = false
)
```

VB

```
Public MustOverride Function GetPropertyVertices (
    value As IComparable,
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
virtual IEnumerable<Vertex^>^ GetPropertyVertices (
    IComparable^ value,
    bool polymorphic = false
) abstract
```

F#

```
abstract GetPropertyVertices :
    value : IComparable *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

value

Type: [System.IComparable](#)

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

VelocityDB Class Library

See Also

[PropertyType Class](#)

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyType.HasPropertyValue Method

Check if an element has a property value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract bool HasPropertyValue (  
    int elementId  
)
```

VB

```
Public MustOverride Function HasPropertyValue (  
    elementId As Integer  
) As Boolean
```

C++

```
public:  
virtual bool HasPropertyValue (  
    int elementId  
) abstract
```

F#

```
abstract HasPropertyValue :  
    elementId : int -> bool
```

Parameters

elementId

Type: [System.Int32](#)

Id of a Vertex/Edge

Return Value

Type: [Boolean](#)

true if element has a property value for this property type

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.RemovePropertyValue Method

Remove a property value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IComparable RemovePropertyValue (
    int elementId
)
```

VB

```
Public MustOverride Function RemovePropertyValue (
    elementId As Integer
) As IComparable
```

C++

```
public:
virtual IComparable^ RemovePropertyValue (
    int elementId
) abstract
```

F#

```
abstract RemovePropertyValue :
    elementId : int -> IComparable
```

Parameters

elementId

Type: [System.Int32](#)

Id of an edge/vertex

Return Value

Type: [IComparable](#)

the value that was assigned prior to removing the property value

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.SetPropertyValue Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------|
|  | SetPropertyValue(EdgeType, Int32, Int32, IComparable) | |
|  | SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element |

See Also

[PropertyType Class](#)

[VelocityGraph Namespace](#)

PropertyType.SetPropertyValue Method (EdgeType, Int32, Int32, IComparable)

[Missing <summary> documentation for

"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void SetPropertyValue (
    EdgeType et,
    int elementId,
    int typeId,
    IComparable value
)
```

VB

```
Public MustOverride Sub SetPropertyValue (
    et As EdgeType,
    elementId As Integer,
    typeId As Integer,
    value As IComparable
)
```

C++

```
public:
virtual void SetPropertyValue (
    EdgeType^ et,
    int elementId,
    int typeId,
    IComparable^ value
) abstract
```

F#

```
abstract SetPropertyValue :
    et : EdgeType *
    elementId : int *
    typeId : int *
    value : IComparable -> unit
```

Parameters

et

Type: [VelocityGraph.EdgeType](#)

[Missing <param name="et"/> documentation for
"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

elementId

Type: [System.Int32](#)

[Missing <param name="elementId"/> documentation for
"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

typeId

Type: [System.Int32](#)

[Missing <param name="typeId"/> documentation for
"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

See Also

[PropertyType Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

PropertyType.SetPropertyValue Method (VertexType, Int32, Int32, IComparable)

Sets a property value for an element

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void SetPropertyValue (
    VertexType vt,
    int elementId,
    int typeId,
    IComparable value
)
```

VB

```
Public MustOverride Sub SetPropertyValue (
    vt As VertexType,
    elementId As Integer,
    typeId As Integer,
    value As IComparable
)
```

C++

```
public:
virtual void SetPropertyValue (
    VertexType^ vt,
    int elementId,
    int typeId,
    IComparable^ value
) abstract
```

F#

```
abstract SetPropertyValue :
    vt : VertexType *
    elementId : int *
    typeId : int *
    value : IComparable -> unit
```

Parameters

vt

Type: [VelocityGraph.VertexType](#)

[Missing <param name="vt"/> documentation for

"M:VelocityGraph.PropertyType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,System.Int32,System.IComparable)"]

VelocityDB Class Library

elementId

Type: [System.Int32](#)

Id of a vertex or edge

typeId

Type: [System.Int32](#)

Id of vertex/edge

value

Type: [System.IComparable](#)

Value to assign to property

See Also

[PropertyType Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T) Class

Used with string property type values. Avoids storing duplicate string values.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityGraph.PropertyType](#)

[VelocityGraph.PropertyTypeT\(UInt64\)](#)

VelocityGraph.PropertyTypeNoDuplicateValues(T)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class PropertyTypeNoDuplicateValues<T> : PropertyTypeT<ulong>
where T : IComparable
```

VB

```
<SerializableAttribute>
Public Class PropertyTypeNoDuplicateValues(Of T As IComparable)
    Inherits PropertyTypeT(Of ULong)
```

C++

```
[SerializableAttribute]
generic<typename T>
where T : IComparable
public ref class PropertyTypeNoDuplicateValues : public
PropertyTypeT<unsigned long long>
```

F#

```
[<SerializableAttribute>]
type PropertyTypeNoDuplicateValues<'T when 'T : IComparable> =
    class
        inherit PropertyTypeT<uint64>
    end
```


Type Parameters

T












always [String](#) for now

The PropertyTypeNoDuplicateValues(T) type exposes the following members.



Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------|
|  | ValueType | Get the value Type (Overrides PropertyTypeT(T).ValueType.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdge(IComparable).) |
|  | GetPropertyEdge(T) | Try to find an Edge with a given property value. |
|  | GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdges(IComparable).) |
|  | GetPropertyEdges(T) | |
|  | GetPropertyValue | Get property value of a Vertex/Edge (Overrides PropertyTypeT(T).GetPropertyValue(Int32).) |
|  | GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyTypeT(T).GetPropertyVertex(IComparable, Boolean, Boolean).) |
|  | GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |
|  | GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyTypeT(T).GetPropertyVertices(IComparable, Boolean).) |
|  | GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |
|  | SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyTypeT(T).SetPropertyValue(EdgeType, Int32, Int32, IComparable).) |
|  | SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyTypeT(T).SetPropertyValue(VertexType, Int32, Int32, IComparable).) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |


See Also

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).PropertyTypeNoDuplicateValues(T) Properties

The [PropertyTypeNoDuplicateValues\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------|
|  | ValueType | Get the value Type (Overrides PropertyTypeT(T).ValueType.) |

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).ValueType Property

Get the value Type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Type ValueType { get; }
```

VB

```
Public Overrides ReadOnly Property ValueType As Type  
    Get
```

C++

```
public:  
virtual property Type^ ValueType {  
    Type^ get () override;  
}
```

F#

```
abstract ValueType : Type with get  
override ValueType : Type with get
```

Property Value

Type: [Type](#)

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)












[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).PropertyTypeNoDuplicateValues(T)



Methods

The [PropertyTypeNoDuplicateValues\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdge(IComparable).) |
|  GetPropertyEdge(T) | Try to find an Edge with a given property value. |
|  GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdges(IComparable).) |
|  GetPropertyEdges(T) | |
|  GetPropertyValue | Get property value of a Vertex/Edge (Overrides PropertyTypeT(T).GetPropertyValue(Int32).) |
|  GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyTypeT(T).GetPropertyVertex(IComparable, Boolean, Boolean).) |
|  GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |
|  GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyTypeT(T).GetPropertyVertices(IComparable, Boolean).) |
|  GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |
|  SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyTypeT(T).SetPropertyValue(EdgeType, Int32, Int32, IComparable).) |
|  SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyTypeT(T).SetPropertyValue(VertexType, Int32, Int32, IComparable).) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |



See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdge Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdge(IComparable) .) |
|  | GetPropertyEdge(T) | Try to find an Edge with a given property value. |

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdge Method (IComparable)

Try to find an [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Edge GetPropertyEdge (
    IComparable value
)
```

VB

```
Public Overrides Function GetPropertyEdge (
    value As IComparable
) As Edge
```

C++

```
public:
virtual Edge^ GetPropertyEdge (
    IComparable^ value
) override
```

F#

```
abstract GetPropertyEdge :
    value : IComparable -> Edge
override GetPropertyEdge :
    value : IComparable -> Edge
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [Edge](#)

An edge with a matching property value

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyEdge Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdge Method (T)

Try to find an [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge GetPropertyEdge (  
    T value  
)
```

VB

```
Public Function GetPropertyEdge (  
    value As T  
) As Edge
```

C++

```
public:  
Edge^ GetPropertyEdge (  
    T value  
)
```

F#

```
member GetPropertyEdge :  
    value : 'T -> Edge
```

Parameters

value

Type: *T*

the property value to look for

Return Value

Type: [Edge](#)

An edge with a matching property value

See Also



[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyEdge Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyTypeT(T).GetPropertyEdges(IComparable) .) |
|  | GetPropertyEdges(T) | |

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdges Method (IComparable)

Try to find all [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<Edge> GetPropertyEdges (
    IComparable value
)
```

VB

```
Public Overrides Function GetPropertyEdges (
    value As IComparable
) As IEnumerable(Of Edge)
```

C++

```
public:
virtual IEnumerable<Edge^>^ GetPropertyEdges (
    IComparable^ value
) override
```

F#

```
abstract GetPropertyEdges :
    value : IComparable -> IEnumerable<Edge>
override GetPropertyEdges :
    value : IComparable -> IEnumerable<Edge>
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [IEnumerable\(Edge\)](#)

An edge with a matching property value

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyEdges Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyEdges Method (T)

[Missing <summary> documentation for

"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyEdges(`0)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Edge> GetPropertyEdges (  
    T value  
)
```

VB

```
Public Function GetPropertyEdges (  
    value As T  
) As IEnumerable(Of Edge)
```

C++

```
public:  
IEnumerable<Edge^>^ GetPropertyEdges (  
    T value  
)
```

F#

```
member GetPropertyEdges :  
    value : 'T -> IEnumerable<Edge>
```

Parameters

value

Type: *T*

[Missing <param name="value"/> documentation for

"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyEdges(`0)"]

Return Value

Type: [IEnumerable\(Edge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyEdges(`0)"]

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[GetPropertyEdges Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyValue Method

Get property value of a Vertex/Edge

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IComparable GetPropertyValue(  
    int element  
)
```

VB

```
Public Overrides Function GetPropertyValue (  
    element As Integer  
) As IComparable
```

C++

```
public:  
virtual IComparable^ GetPropertyValue(  
    int element  
) override
```

F#

```
abstract GetPropertyValue :  
    element : int -> IComparable  
override GetPropertyValue :  
    element : int -> IComparable
```

Parameters

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyValue(System.Int32)"]

Return Value

Type: [IComparable](#)

the property value



See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyVertex Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  | GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyTypeT(T).GetPropertyVertex(IComparable, Boolean, Boolean) .) |
|  | GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyVertex Method (IComparable, Boolean, Boolean)

[Missing <summary> documentation for "M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Vertex GetPropertyVertex(
    IComparable value,
    bool polymorphic = false,
    bool errorIfNotFound = true
)
```

VB

```
Public Overrides Function GetPropertyVertex (
    value As IComparable,
    Optional polymorphic As Boolean = false,
    Optional errorIfNotFound As Boolean = true
) As Vertex
```

C++

```
public:
virtual Vertex^ GetPropertyVertex(
    IComparable^ value,
    bool polymorphic = false,
    bool errorIfNotFound = true
) override
```

F#

```
abstract GetPropertyVertex :
    value : IComparable *
    ?polymorphic : bool *
    ?errorIfNotFound : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
    let_errorIfNotFound = defaultArg errorIfNotFound true
*)
-> Vertex
override GetPropertyVertex :
    value : IComparable *
    ?polymorphic : bool *
    ?errorIfNotFound : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
    let_errorIfNotFound = defaultArg errorIfNotFound true
```

```
*)  
-> Vertex
```

Parameters

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyVertex(System.IComparable,System.
.Boolean,System.Boolean)"]

polymorphic (Optional)

Type: [System.Boolean](#)

[Missing <param name="polymorphic"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyVertex(System.IComparable,System.
.Boolean,System.Boolean)"]

errorIfNotFound (Optional)

Type: [System.Boolean](#)

[Missing <param name="errorIfNotFound"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyVertex(System.IComparable,System.
.Boolean,System.Boolean)"]

Return Value

Type: [Vertex](#)

[Missing <returns> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.GetPropertyVertex(System.IComparable,System.
.Boolean,System.Boolean)"]

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyVertex Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyVertex Method (T, Boolean, Boolean)

Try to find a [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex GetPropertyVertex(  
    T value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Function GetPropertyVertex (  
    value As T,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
Vertex^ GetPropertyVertex(  
    T value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

F#

```
member GetPropertyVertex :  
    value : 'T *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
    let_errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex
```

Parameters

value

Type: *T*

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

errorIfNotFound (Optional)

Type: [System.Boolean](#)

If true, signal an error if no matching [Vertex](#) found

Return Value

Type: [Vertex](#)

A matching Vertex

See Also



[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyVertex Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyTypeT(T).GetPropertyVertices(IComparable, Boolean) .) |
|  | GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).GetPropertyVertices Method (IComparable, Boolean)

Try to find all [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<Vertex> GetPropertyVertices (
    IComparable value,
    bool polymorphic = false
)
```

VB

```
Public Overrides Function GetPropertyVertices (
    value As IComparable,
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
virtual IEnumerable<Vertex^>^ GetPropertyVertices (
    IComparable^ value,
    bool polymorphic = false
) override
```

F#

```
abstract GetPropertyVertices :
    value : IComparable *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
override GetPropertyVertices :
    value : IComparable *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

value

Type: [System.IComparable](#)

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(*T*).GetPropertyVertices Method (*T*, Boolean)

Try to find all [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Vertex> GetPropertyVertices (
    T value,
    bool polymorphic = false
)
```

VB

```
Public Function GetPropertyVertices (
    value As T,
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
    IEnumerable<Vertex^>^ GetPropertyVertices (
        T value,
        bool polymorphic = false
    )
```

F#

```
member GetPropertyVertices :
    value : 'T *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

value

Type: *T*

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

VelocityDB Class Library

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

See Also



[PropertyTypeNoDuplicateValues\(T\)Class](#)

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).SetPropertyValue Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|  | SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyTypeT(T).SetPropertyValue(EdgeType, Int32, Int32, IComparable) .) |
|  | SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyTypeT(T).SetPropertyValue(VertexType, Int32, Int32, IComparable) .) |

See Also

[PropertyTypeNoDuplicateValues\(T\) Class](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).SetPropertyValue Method (EdgeType, Int32, Int32, IComparable)

[Missing <summary> documentation for "M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetPropertyValue(  
    EdgeType et,  
    int element,  
    int typeId,  
    IComparable value  
)
```

VB

```
Public Overrides Sub SetPropertyValue (  
    et As EdgeType,  
    element As Integer,  
    typeId As Integer,  
    value As IComparable  
)
```

C++

```
public:  
virtual void SetPropertyValue(  
    EdgeType^ et,  
    int element,  
    int typeId,  
    IComparable^ value  
) override
```

F#

```
abstract SetPropertyValue :  
    et : EdgeType *  
    element : int *  
    typeId : int *  
    value : IComparable -> unit  
override SetPropertyValue :  
    et : EdgeType *  
    element : int *  
    typeId : int *  
    value : IComparable -> unit
```

Parameters

et

Type: [VelocityGraph.EdgeType](#)

[Missing <param name="et"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

typed

Type: [System.Int32](#)

[Missing <param name="typed"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeNoDuplicateValues(T).SetPropertyValue Method (VertexType, Int32, Int32, IComparable)

Sets a property value for an element

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetPropertyValue (
    VertexType vt,
    int element,
    int typeId,
    IComparable value
)
```

VB

```
Public Overrides Sub SetPropertyValue (
    vt As VertexType,
    element As Integer,
    typeId As Integer,
    value As IComparable
)
```

C++

```
public:
virtual void SetPropertyValue (
    VertexType^ vt,
    int element,
    int typeId,
    IComparable^ value
) override
```

F#

```
abstract SetPropertyValue :
    vt : VertexType *
    element : int *
    typeId : int *
    value : IComparable -> unit
override SetPropertyValue :
    vt : VertexType *
    element : int *
    typeId : int *
    value : IComparable -> unit
```

Parameters

vt

Type: [VelocityGraph.VertexType](#)

[Missing <param name="vt"/> documentation for "M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.VertexType,System.Int32,System.Int32,System.IComparable)"]

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.PropertyTypeNoDuplicateValues`1.SetPropertyValue(VelocityGraph.VertexType,System.Int32,System.Int32,System.IComparable)"]

typeId

Type: [System.Int32](#)

Id of vertex/edge

value

Type: [System.IComparable](#)

Value to assign to property

See Also

[PropertyTypeNoDuplicateValues\(T\)Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T) Class

Keeper of [Edge](#) and [Vertex](#) properties

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

[VelocityGraph.PropertyType](#)

VelocityGraph.PropertyTypeT(T)

[VelocityGraph.PropertyTypeNoDuplicateValues\(T\)](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PropertyTypeT<T> : PropertyType  
where T : IComparable
```

VB

```
<SerializableAttribute>  
Public Class PropertyTypeT(Of T As IComparable)  
    Inherits PropertyType
```

C++

```
[SerializableAttribute]  
generic<typename T>  
where T : IComparable  
public ref class PropertyTypeT : public PropertyType
```

F#

```
[<SerializableAttribute>]  
type PropertyTypeT<'T when 'T : IComparable> =  
    class  
        inherit PropertyType  
    end
```


Type Parameters

T















The type of value used by the property type

The PropertyTypeT(T) type exposes the following members.


Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------|
|  | ValueType | Get the value Type (Overrides PropertyType.ValueType .) |


Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyType.GetPropertyEdge(IComparable) .) |
|  | GetPropertyEdge(T) | Try to find an Edge with a given property value. |
|  | GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyType.GetPropertyEdges(IComparable) .) |
|  | GetPropertyEdges(T) | |
|  | GetPropertyValue | Get property value of a Vertex/Edge (Overrides PropertyType.GetPropertyValue(Int32) .) |
|  | GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyType.GetPropertyVertex(IComparable, Boolean, Boolean) .) |
|  | GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |
|  | GetPropertyVertices(Boolean) | Try to find all Vertex with a property value. (Overrides PropertyType.GetPropertyVertices(Boolean) .) |
|  | GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyType.GetPropertyVertices(IComparable, Boolean) .) |
|  | GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |
|  | HasPropertyValue | Does a certain element (Edge/Vertex) have a property value for this property type (Overrides PropertyType.HasPropertyValue(Int32) .) |
|  | RemovePropertyValue | Remove a property value (Overrides PropertyType.RemovePropertyValue(Int32) .) |
|  | SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyType.SetPropertyValue(EdgeType, Int32, Int32, IComparable) .) |
|  | SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyType.SetPropertyValue(VertexType, Int32, Int32, IComparable) .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|


See Also

[VelocityGraph Namespace](#)

PropertyTypeT(T).PropertyTypeT(T) Properties

The [PropertyTypeT\(T\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------|
|  | ValueType | Get the value Type (Overrides PropertyType.ValueType.) |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).ValueType Property

Get the value Type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Type ValueType { get; }
```

VB

```
Public Overrides ReadOnly Property ValueType As Type  
    Get
```

C++

```
public:  
virtual property Type^ ValueType {  
    Type^ get () override;  
}
```

F#

```
abstract ValueType : Type with get  
override ValueType : Type with get
```

Property Value

Type: [Type](#)

See Also















[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)



PropertyTypeT(T).PropertyTypeT(T) Methods

The [PropertyTypeT\(T\)](#) generic type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyType.GetPropertyEdge(IComparable) .) |
|  GetPropertyEdge(T) | Try to find an Edge with a given property value. |
|  GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyType.GetPropertyEdges(IComparable) .) |
|  GetPropertyEdges(T) | |
|  GetPropertyValue | Get property value of a Vertex/Edge (Overrides PropertyType.GetPropertyValue(Int32) .) |
|  GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyType.GetPropertyVertex(IComparable, Boolean, Boolean) .) |
|  GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |
|  GetPropertyVertices(Boolean) | Try to find all Vertex with a property value. (Overrides PropertyType.GetPropertyVertices(Boolean) .) |
|  GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyType.GetPropertyVertices(IComparable, Boolean) .) |
|  GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |
|  HasPropertyValue | Does a certain element (Edge/Vertex) have a property value for this property type (Overrides PropertyType.HasPropertyValue(Int32) .) |
|  RemovePropertyValue | Remove a property value (Overrides PropertyType.RemovePropertyValue(Int32) .) |
|  SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyType.SetPropertyValue(EdgeType, Int32, Int32, IComparable) .) |
|  SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyType.SetPropertyValue(VertexType, Int32, Int32, IComparable) .) |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

VelocityDB Class Library



See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdge Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdge(IComparable) | Try to find an Edge with a given property value. (Overrides PropertyType.GetPropertyEdge(IComparable) .) |
|  | GetPropertyEdge(T) | Try to find an Edge with a given property value. |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdge Method (IComparable)

Try to find an [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Edge GetPropertyEdge (
    IComparable value
)
```

VB

```
Public Overrides Function GetPropertyEdge (
    value As IComparable
) As Edge
```

C++

```
public:
virtual Edge^ GetPropertyEdge (
    IComparable^ value
) override
```

F#

```
abstract GetPropertyEdge :
    value : IComparable -> Edge
override GetPropertyEdge :
    value : IComparable -> Edge
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [Edge](#)

An edge with a matching property value

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyEdge Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdge Method (T)

Try to find an [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge GetPropertyEdge (  
    T value  
)
```

VB

```
Public Function GetPropertyEdge (  
    value As T  
) As Edge
```

C++

```
public:  
Edge^ GetPropertyEdge (  
    T value  
)
```

F#

```
member GetPropertyEdge :  
    value : 'T -> Edge
```

Parameters

value

Type: *T*

the property value to look for

Return Value

Type: [Edge](#)

An edge with a matching property value

See Also



[PropertyTypeT\(T\)Class](#)

[GetPropertyEdge Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyEdges(IComparable) | Try to find all Edge with a given property value. (Overrides PropertyType.GetPropertyEdges(IComparable) .) |
|  | GetPropertyEdges(T) | |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdges Method (IComparable)

Try to find all [Edge](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<Edge> GetPropertyEdges (
    IComparable value
)
```

VB

```
Public Overrides Function GetPropertyEdges (
    value As IComparable
) As IEnumerable(Of Edge)
```

C++

```
public:
virtual IEnumerable<Edge^>^ GetPropertyEdges (
    IComparable^ value
) override
```

F#

```
abstract GetPropertyEdges :
    value : IComparable -> IEnumerable<Edge>
override GetPropertyEdges :
    value : IComparable -> IEnumerable<Edge>
```

Parameters

value

Type: [System.IComparable](#)

the property value to look for

Return Value

Type: [IEnumerable\(Edge\)](#)

An edge with a matching property value

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyEdges Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyEdges Method (T)

[Missing <summary> documentation for "M:VelocityGraph.PropertyType`1.GetPropertyEdges(`0)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Edge> GetPropertyEdges (
    T value
)
```

VB

```
Public Function GetPropertyEdges (
    value As T
) As IEnumerable(Of Edge)
```

C++

```
public:
    IEnumerable<Edge^>^ GetPropertyEdges (
        T value
    )
```

F#

```
member GetPropertyEdges :
    value : 'T -> IEnumerable<Edge>
```

Parameters

value

Type: *T*

[Missing <param name="value"/> documentation for "M:VelocityGraph.PropertyType`1.GetPropertyEdges(`0)"]

Return Value

Type: [IEnumerable\(Edge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.PropertyType`1.GetPropertyEdges(`0)"]

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyEdges Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyValue Method

Get property value of a Vertex/Edge

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IComparable GetPropertyValue(  
    int element  
)
```

VB

```
Public Overrides Function GetPropertyValue (  
    element As Integer  
) As IComparable
```

C++

```
public:  
virtual IComparable^ GetPropertyValue(  
    int element  
) override
```

F#

```
abstract GetPropertyValue :  
    element : int -> IComparable  
override GetPropertyValue :  
    element : int -> IComparable
```

Parameters

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.PropertyType`1.GetPropertyValue(System.Int32)"]

Return Value

Type: [IComparable](#)

the property value



See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertex Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
|  | GetPropertyVertex(IComparable, Boolean, Boolean) | (Overrides PropertyType.GetPropertyVertex(IComparable, Boolean, Boolean) .) |
|  | GetPropertyVertex(T, Boolean, Boolean) | Try to find a Vertex with a given property value. |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertex Method (IComparable, Boolean, Boolean)

[Missing <summary> documentation for "M:VelocityGraph.PropertyTypeT`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Vertex GetPropertyVertex(  
    IComparable value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Overrides Function GetPropertyVertex (  
    value As IComparable,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
virtual Vertex^ GetPropertyVertex(  
    IComparable^ value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
) override
```

F#

```
abstract GetPropertyVertex :  
    value : IComparable *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
    let_errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex  
override GetPropertyVertex :  
    value : IComparable *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
    let_errorIfNotFound = defaultArg errorIfNotFound true
```

```
*)  
-> Vertex
```

Parameters

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.PropertyTypeT`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

polymorphic (Optional)

Type: [System.Boolean](#)

[Missing <param name="polymorphic"/> documentation for "M:VelocityGraph.PropertyTypeT`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

errorIfNotFound (Optional)

Type: [System.Boolean](#)

[Missing <param name="errorIfNotFound"/> documentation for "M:VelocityGraph.PropertyTypeT`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

Return Value

Type: [Vertex](#)

[Missing <returns> documentation for "M:VelocityGraph.PropertyTypeT`1.GetPropertyVertex(System.IComparable,System.Boolean,System.Boolean)"]

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyVertex Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertex Method (T, Boolean, Boolean)

Try to find a [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex GetPropertyVertex(  
    T value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Function GetPropertyVertex (  
    value As T,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
Vertex^ GetPropertyVertex(  
    T value,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

F#

```
member GetPropertyVertex :  
    value : 'T *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let _polymorphic = defaultArg polymorphic false  
    let _errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex
```

Parameters

value

Type: *T*

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

VelocityDB Class Library

If true, also look for property value matching vertices of property [VertexType](#) sub classes

errorIfNotFound (Optional)

Type: [System.Boolean](#)

If true, signal an error if no matching [Vertex](#) found

Return Value

Type: [Vertex](#)

A matching Vertex

See Also




[PropertyTypeT\(T\)Class](#)

[GetPropertyVertex Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetPropertyVertices(Boolean) | Try to find all Vertex with a property value. (Overrides PropertyType.GetPropertyVertices(Boolean) .) |
|  | GetPropertyVertices(IComparable, Boolean) | Try to find all Vertex with a given property value. (Overrides PropertyType.GetPropertyVertices(IComparable, Boolean) .) |
|  | GetPropertyVertices(T, Boolean) | Try to find all Vertex with a given property value. |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertices Method (Boolean)

Try to find all [Vertex](#) with a property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<Vertex> GetPropertyVertices (
    bool polymorphic = false
)
```

VB

```
Public Overrides Function GetPropertyVertices (
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
virtual IEnumerable<Vertex^>^ GetPropertyVertices (
    bool polymorphic = false
) override
```

F#

```
abstract GetPropertyVertices :
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
override GetPropertyVertices :
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

See Also

[PropertyTypeT\(T\)Class](#)

VelocityDB Class Library

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertices Method (IComparable, Boolean)

Try to find all [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<Vertex> GetPropertyVertices (
    IComparable value,
    bool polymorphic = false
)
```

VB

```
Public Overrides Function GetPropertyVertices (
    value As IComparable,
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
virtual IEnumerable<Vertex^>^ GetPropertyVertices (
    IComparable^ value,
    bool polymorphic = false
) override
```

F#

```
abstract GetPropertyVertices :
    value : IComparable *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
override GetPropertyVertices :
    value : IComparable *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

value

Type: [System.IComparable](#)

The property value to look for

polymorphic (Optional)

VelocityDB Class Library

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).GetPropertyVertices Method (T, Boolean)

Try to find all [Vertex](#) with a given property value.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Vertex> GetPropertyVertices (
    T value,
    bool polymorphic = false
)
```

VB

```
Public Function GetPropertyVertices (
    value As T,
    Optional polymorphic As Boolean = false
) As IEnumerable(Of Vertex)
```

C++

```
public:
    IEnumerable<Vertex^>^ GetPropertyVertices (
        T value,
        bool polymorphic = false
    )
```

F#

```
member GetPropertyVertices :
    value : 'T *
    ?polymorphic : bool
(* Defaults:
    let_polymorphic = defaultArg polymorphic false
*)
-> IEnumerable<Vertex>
```

Parameters

value

Type: *T*

The property value to look for

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also look for property value matching vertices of property [VertexType](#) sub classes

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of matching vertices

VelocityDB Class Library

See Also

[PropertyTypeT\(T\)Class](#)

[GetPropertyVertices Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).HasPropertyValue Method

Does a certain element (Edge/Vertex) have a property value for this property type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool HasPropertyValue(  
    int elementId  
)
```

VB

```
Public Overrides Function HasPropertyValue (  
    elementId As Integer  
) As Boolean
```

C++

```
public:  
virtual bool HasPropertyValue(  
    int elementId  
) override
```

F#

```
abstract HasPropertyValue :  
    elementId : int -> bool  
override HasPropertyValue :  
    elementId : int -> bool
```

Parameters

elementId

Type: [System.Int32](#)

element id of element

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.PropertyType`1.HasPropertyValue(System.Int32)"]

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).RemovePropertyValue Method

Remove a property value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IComparable RemovePropertyValue (
    int element
)
```

VB

```
Public Overrides Function RemovePropertyValue (
    element As Integer
) As IComparable
```

C++

```
public:
virtual IComparable^ RemovePropertyValue (
    int element
) override
```

F#

```
abstract RemovePropertyValue :
    element : int -> IComparable
override RemovePropertyValue :
    element : int -> IComparable
```

Parameters

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.PropertyType`1.RemovePropertyValue(System.Int32)"]

Return Value

Type: [IComparable](#)

the value that was assigned prior to removing the property value



See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).SetPropertyValue Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
|  | SetPropertyValue(EdgeType, Int32, Int32, IComparable) | (Overrides PropertyType.SetPropertyValue(EdgeType, Int32, Int32, IComparable) .) |
|  | SetPropertyValue(VertexType, Int32, Int32, IComparable) | Sets a property value for an element (Overrides PropertyType.SetPropertyValue(VertexType, Int32, Int32, IComparable) .) |

See Also

[PropertyTypeT\(T\)Class](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).SetPropertyValue Method (EdgeType, Int32, Int32, IComparable)

[Missing <summary> documentation for "M:VelocityGraph.PropertyTypeT`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetPropertyValue (
    EdgeType et,
    int element,
    int typeId,
    IComparable value
)
```

VB

```
Public Overrides Sub SetPropertyValue (
    et As EdgeType,
    element As Integer,
    typeId As Integer,
    value As IComparable
)
```

C++

```
public:
virtual void SetPropertyValue (
    EdgeType^ et,
    int element,
    int typeId,
    IComparable^ value
) override
```

F#

```
abstract SetPropertyValue :
    et : EdgeType *
    element : int *
    typeId : int *
    value : IComparable -> unit
override SetPropertyValue :
    et : EdgeType *
    element : int *
    typeId : int *
    value : IComparable -> unit
```

Parameters

et

Type: [VelocityGraph.EdgeType](#)

[Missing <param name="et"/> documentation for "M:VelocityGraph.PropertyTypeT`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.PropertyTypeT`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

typeId

Type: [System.Int32](#)

[Missing <param name="typeId"/> documentation for "M:VelocityGraph.PropertyTypeT`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

value

Type: [System.IComparable](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.PropertyTypeT`1.SetPropertyValue(VelocityGraph.EdgeType,System.Int32,System.Int32,System.IComparable)"]

See Also

[PropertyTypeT\(T\)Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

PropertyTypeT(T).SetPropertyValue Method (VertexType, Int32, Int32, IComparable)

Sets a property value for an element

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetPropertyValue (
    VertexType vt,
    int element,
    int typeId,
    IComparable value
)
```

VB

```
Public Overrides Sub SetPropertyValue (
    vt As VertexType,
    element As Integer,
    typeId As Integer,
    value As IComparable
)
```

C++

```
public:
virtual void SetPropertyValue (
    VertexType^ vt,
    int element,
    int typeId,
    IComparable^ value
) override
```

F#

```
abstract SetPropertyValue :
    vt : VertexType *
    element : int *
    typeId : int *
    value : IComparable -> unit
override SetPropertyValue :
    vt : VertexType *
    element : int *
    typeId : int *
    value : IComparable -> unit
```

Parameters

vt

Type: [VelocityGraph.VertexType](#)

[Missing <param name="vt"/> documentation for
"M:VelocityGraph.PropertyType`1.SetPropertyValue(VelocityGraph.VertexType,System.Int32,System.Int32,System.IComparable)"]

element

Type: [System.Int32](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.PropertyType`1.SetPropertyValue(VelocityGraph.VertexType,System.Int32,System.Int32,System.IComparable)"]

typeId

Type: [System.Int32](#)

Id of vertex/edge

value

Type: [System.IComparable](#)

Value to assign to property

See Also

[PropertyType\(T\)Class](#)

[SetPropertyValue Overload](#)

[VelocityGraph Namespace](#)

Vertex Class

A vertex maintains pointers to both a set of incoming and outgoing edges. The outgoing edges are those edges for which the vertex is the tail. The incoming edges are those edges for which the vertex is the head. Diagrammatically, ---inEdges---> vertex ---outEdges---

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Element](#)

VelocityGraph.Vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Vertex : Element, IVertex,
    IElement, IDictionary<string, Object>, ICollection<KeyValuePair<string,
Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection, IEqualityComparer<Vertex>
```

VB

```
Public Class Vertex
    Inherits Element
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection, IEqualityComparer(Of Vertex)
```

C++

```
public ref class Vertex : public Element,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection, IEqualityComparer<Vertex^>
```

F#

```
type Vertex =
    class
        inherit Element
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
```


```

interface IDictionary
interface ICollection
interface IEqualityComparer<Vertex>
end




```

The **Vertex** type exposes the following members.










Constructors














| Name | Description |
|----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Vertex | Normally you should use GetVertex(Int32, Boolean, Boolean) but if you need a reference to a Vertex that has no yet been created, this constructor may be used (but know what you are doing!) |

Properties

| Name | Description |
|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. (Overrides DictionaryElement.Id.) |
|  VertexId | Gets the unique id of a vertex |
|  VertexType | Gets the vertex type of this Vertex |












Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge(String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |
|  AddEdge(EdgeType, Vertex) | Add an edge from this Vertex to inVertex of edge type. |
|  AddEdge(Object, String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |
|  Equals(Object) | Determines whether the specified Vertex is equal to the current one. (Overrides Object.Equals(Object).) |
|  Equals(Vertex, Vertex) | Determines whether the specified objects are equal. |
|  GetEdges(Direction, String[]) | Return the edges incident to the vertex according to the provided direction and edge labels. |
|  GetEdges(EdgeType, Direction) | Selects all edges from or to this vertex and for the given edge type. |
|  GetHashCode() | Use id as hash code (Overrides Element.GetHashCode().) |
|  GetHashCode(Vertex) | Returns a hash code for the specified object. |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetNumberOfEdges(EdgeType, Direction) | Gets the number of edges from or to this vertex and for the given edge type. |
|  GetNumberOfEdges(EdgeType, Vertex, Direction) | Gets the number of edges from or to this vertex for the given edge type and the given other Vertex. |
|  GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String) .) |
|  GetProperty(PropertyType) | Gets the Value for the given Property id |
|  GetPropertyKeys | Return all the keys associated with the vertex. (Overrides DictionaryElement.GetPropertyKeys() .) |
|  GetVertices | Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge). |
|  Query | Uses DefaultVertexQuery |
|  RelatedVertices | |
|  Remove | Removes this Vertex from the Graph (Overrides DictionaryElement.Remove() .) |
|  RemoveProperty | Un-assigns a key/value property from the vertex. The object value of the removed property is returned. (Overrides DictionaryElement.RemoveProperty(String) .) |
|  SetProperty(String, Object) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  SetProperty(PropertyType, IComparable) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |
|  Traverse(Direction, ISet(EdgeType)) | Selects all neighbor Vertices from or to this vertex and for the given edge types. |
|  Traverse(EdgeType, Direction) | Selects all neighbor Vertices from or to this vertex and for the given edge type. |
|  Traverse(Int32, Boolean, Direction, Vertex, ISet(EdgeType), ISet(VertexType), ISet(VertexType), ISet(Vertex), ISet(Vertex), ISet(Edge), ISet(Edge), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), Func(Vertex, Boolean), | Traverses graph from this Vertex to a target Vertex using Breadth-first search like in Dijkstra's algorithm |

| | |
|-----------------------------------------------------------------|--|
| Func(Edge, Boolean), Func(List(Edge), Boolean)) | |
|-----------------------------------------------------------------|--|

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |

See Also

[VelocityGraph Namespace](#)

Vertex Constructor

Normally you should use [GetVertex\(Int32, Boolean, Boolean\)](#) but if you need a reference to a Vertex that has not yet been created, this constructor may be used (but know what you are doing!)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex(  
    Graph g,  
    VertexType eType,  
    int eId  
)
```

VB

```
Public Sub New (  
    g As Graph,  
    eType As VertexType,  
    eId As Integer  
)
```

C++

```
public:  
Vertex(  
    Graph^ g,  
    VertexType^ eType,  
    int eId  
)
```

F#

```
new :  
    g : Graph *  
    eType : VertexType *  
    eId : int -> Vertex
```

Parameters

g

Type: [VelocityGraph.Graph](#)

the owning graph

eType

Type: [VelocityGraph.VertexType](#)

the type of the Vertex

eId

Type: [System.Int32](#)

the Id of the Vertex

See Also




[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Vertex Properties

The [Vertex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. (Overrides DictionaryElement.Id.) |
|  | VertexId | Gets the unique id of a vertex |
|  | VertexType | Gets the vertex type of this Vertex |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Id Property

An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Return Value

Type: [Object](#)

the identifier of the element

Implements

[IElement.Id](#)

[IElement.Id](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.VertexId Property

Gets the unique id of a vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int VertexId { get; }
```

VB

```
Public ReadOnly Property VertexId As Integer  
    Get
```

C++

```
public:  
property int VertexId {  
    int get ();  
}
```

F#

```
member VertexId : int with get
```

Property Value

Type: [Int32](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.VertexType Property

Gets the vertex type of this Vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexType VertexType { get; }
```

VB

```
Public ReadOnly Property VertexType As VertexType  
    Get
```

C++

```
public:  
property VertexType^ VertexType {  
    VertexType^ get ();  
}
```

F#

```
member VertexType : VertexType with get
```

Property Value

Type: [VertexType](#)

See Also


















[Vertex Class](#)









[VelocityGraph Namespace](#)

Vertex.Vertex Methods




The [Vertex](#) type exposes the following members.









Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge(String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |
|  AddEdge(EdgeType, Vertex) | Add an edge from this Vertex to inVertex of edge type. |
|  AddEdge(Object, String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |
|  Equals(Object) | Determines whether the specified Vertex is equal to the current one. (Overrides Object.Equals(Object) .) |
|  Equals(Vertex, Vertex) | Determines whether the specified objects are equal. |
|  GetEdges(Direction, String[]) | Return the edges incident to the vertex according to the provided direction and edge labels. |
|  GetEdges(EdgeType, Direction) | Selects all edges from or to this vertex and for the given edge type. |
|  GetHashCode() | Use id as hash code (Overrides Element.GetHashCode() .) |
|  GetHashCode(Vertex) | Returns a hash code for the specified object. |
|  GetNumberOfEdges(EdgeType, Direction) | Gets the number of edges from or to this vertex and for the given edge type. |
|  GetNumberOfEdges(EdgeType, Vertex, Direction) | Gets the number of edges from or to this vertex for the given edge type and the given other Vertex. |
|  GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String) .) |
|  GetProperty(PropertyType) | Gets the Value for the given Property id |
|  GetPropertyKeys | Return all the keys associated with the vertex. (Overrides DictionaryElement.GetPropertyKeys() .) |
|  GetVertices | Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge). |
|  Query | Uses DefaultVertexQuery |
|  RelatedVertices | |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Remove | Removes this Vertex from the Graph (Overrides DictionaryElement.Remove() .) |
|  RemoveProperty | Un-assigns a key/value property from the vertex. The object value of the removed property is returned. (Overrides DictionaryElement.RemoveProperty(String) .) |
|  SetProperty(String, Object) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  SetProperty(PropertyType, IComparable) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. |
|  ToString | Returns a string that represents the current object. (Overrides Object.ToString() .) |
|  Traverse(Direction, ISet(EdgeType)) | Selects all neighbor Vertices from or to this vertex and for the given edge types. |
|  Traverse(EdgeType, Direction) | Selects all neighbor Vertices from or to this vertex and for the given edge type. |
|  Traverse(Int32, Boolean, Direction, Vertex, ISet(EdgeType), ISet(VertexType), ISet(VertexType), ISet(Vertex), ISet(Vertex), ISet(Edge), ISet(Edge), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), Func(Vertex, Boolean), Func(Edge, Boolean), Func(List(Edge), Boolean)) | Traverses graph from this Vertex to a target Vertex using Breadth-first search like in Dijkstra's algorithm |

Extension Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |




See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.AddEdge Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge(String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |
|  | AddEdge(EdgeType, Vertex) | Add an edge from this Vertex to inVertex of edge type. |
|  | AddEdge(Object, String, IVertex) | Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.AddEdge Method (String, IVertex)

Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge(  
    string label,  
    IVertex inVertex  
)
```

VB

```
Public Function AddEdge (  
    label As String,  
    inVertex As IVertex  
) As IEdge
```

C++

```
public:  
IEdge^ AddEdge(  
    String^ label,  
    IVertex^ inVertex  
)
```

F#

```
member AddEdge :  
    label : string *  
    inVertex : IVertex -> IEdge
```

Parameters

label

Type: [System.String](#)

The type of edge to create

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

The head of the new edge

Return Value

Type: [IEdge](#)

the new edge

VelocityDB Class Library

See Also

[Vertex Class](#)

[AddEdge Overload](#)

[VelocityGraph Namespace](#)

Vertex.AddEdge Method (EdgeType, Vertex)

Add an edge from this Vertex to inVertex of edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Edge AddEdge(  
    EdgeType edgeType,  
    Vertex head  
)
```

VB

```
Public Function AddEdge (  
    edgeType As EdgeType,  
    head As Vertex  
) As Edge
```

C++

```
public:  
Edge^ AddEdge (  
    EdgeType^ edgeType,  
    Vertex^ head  
)
```

F#

```
member AddEdge :  
    edgeType : EdgeType *  
    head : Vertex -> Edge
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

The type of edge to add

head

Type: [VelocityGraph.Vertex](#)

The head of the new edge

Return Value

Type: [Edge](#)

the new edge

See Also

[Vertex Class](#)

VelocityDB Class Library

[AddEdge Overload](#)

[VelocityGraph Namespace](#)

Vertex.AddEdge Method (Object, String, IVertex)

Add an edge from this Vertex to inVertex of edge type looked up from label, if edge type does not yet exist it is created.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge(  
    Object edgeId,  
    string label,  
    IVertex inVertex  
)
```

VB

```
Public Function AddEdge (  
    edgeId As Object,  
    label As String,  
    inVertex As IVertex  
) As IEdge
```

C++

```
public:  
virtual IEdge^ AddEdge(  
    Object^ edgeId,  
    String^ label,  
    IVertex^ inVertex  
) sealed
```

F#

```
abstract AddEdge :  
    edgeId : Object *  
    label : string *  
    inVertex : IVertex -> IEdge  
override AddEdge :  
    edgeId : Object *  
    label : string *  
    inVertex : IVertex -> IEdge
```

Parameters

edgeId

Type: [System.Object](#)

If not null, this must be a UInt32 to be used as edge id - NOTE: not yet implemented usage

label

Type: [System.String](#)

The type of edge to create

VelocityDB Class Library

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

The head of the new edge

Return Value

Type: [IEdge](#)

the new edge

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also



[Vertex Class](#)

[AddEdge Overload](#)

[VelocityGraph Namespace](#)

Vertex.Equals Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
|  | Equals(Object) | Determines whether the specified Vertex is equal to the current one. (Overrides Object.Equals(Object) .) |
|  | Equals(Vertex, Vertex) | Determines whether the specified objects are equal. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Equals Method (Object)

Determines whether the specified [Vertex](#) is equal to the current one.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object other  
)
```

VB

```
Public Overrides Function Equals (  
    other As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ other  
) override
```

F#

```
abstract Equals :  
    other : Object -> bool  
override Equals :  
    other : Object -> bool
```

Parameters

other

Type: [System.Object](#)

The object to compare with the current object.

Return Value

Type: [Boolean](#)

`true` if the specified object is equal to the current object; otherwise, `false`.

See Also

[Vertex Class](#)

[Equals Overload](#)

[VelocityGraph Namespace](#)

Vertex.Equals Method (Vertex, Vertex)

Determines whether the specified objects are equal.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Equals(  
    Vertex x,  
    Vertex y  
)
```

VB

```
Public Function Equals (  
    x As Vertex,  
    y As Vertex  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Vertex^ x,  
    Vertex^ y  
) sealed
```

F#

```
abstract Equals :  
    x : Vertex *  
    y : Vertex -> bool  
override Equals :  
    x : Vertex *  
    y : Vertex -> bool
```

Parameters

x

Type: [VelocityGraph.Vertex](#)

The first object of type *T* to compare.

y

Type: [VelocityGraph.Vertex](#)

The second object of type *T* to compare.

Return Value

Type: [Boolean](#)

`true` (True in Visual Basic) if the specified objects are equal; otherwise, `false` (False in Visual Basic).

VelocityDB Class Library

Implements

[IEqualityComparer\(T\).Equals\(T, T\)](#)

See Also



[Vertex Class](#)

[Equals Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------------------------------------------------------|
|  | GetEdges(Direction,String[]) | Return the edges incident to the vertex according to the provided direction and edge labels. |
|  | GetEdges(EdgeType,Direction) | Selects all edges from or to this vertex and for the given edge type. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.GetEdges Method (Direction, String[])

Return the edges incident to the vertex according to the provided direction and edge labels.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

the direction of the edges to retrieve

labels

Type: [System.String](#)[]

the labels of the edges to retrieve

Return Value

Type: [IEnumerable\(IEdge\)](#)

an IEnumerable of incident edges

VelocityDB Class Library

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[Vertex Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetEdges Method (EdgeType, Direction)

Selects all edges from or to this vertex and for the given edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    EdgeType edgeType,  
    Direction dir  
)
```

VB

```
Public Function GetEdges (  
    edgeType As EdgeType,  
    dir As Direction  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
IEnumerable<IEdge^>^ GetEdges (  
    EdgeType^ edgeType,  
    Direction dir  
)
```

F#

```
member GetEdges :  
    edgeType : EdgeType *  
    dir : Direction -> IEnumerable<IEdge>
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

the id of an EdgeType

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

direction, one of: Out, In, Both

Return Value

Type: [IEnumerable\(IEdge\)](#)

a set of Edge

See Also

[Vertex Class](#)



VelocityDB Class Library

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetHashCode Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-------------------------------------------------------------------------|
|  | GetHashCode() | Use id as hash code (Overrides Element.GetHashCode() .) |
|  | GetHashCode(Vertex) | Returns a hash code for the specified object. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.GetHashCode Method

Use id as hash code

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

The hash code given by id

See Also

[Vertex Class](#)

[GetHashCode Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetHashCode Method (Vertex)

Returns a hash code for the specified object.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int GetHashCode(  
    Vertex obj  
)
```

VB

```
Public Function GetHashCode (  
    obj As Vertex  
) As Integer
```

C++

```
public:  
virtual int GetHashCode(  
    Vertex^ obj  
) sealed
```

F#

```
abstract GetHashCode :  
    obj : Vertex -> int  
override GetHashCode :  
    obj : Vertex -> int
```

Parameters

obj

Type: [VelocityGraph.Vertex](#)

The [Object](#) for which a hash code is to be returned.

Return Value

Type: [Int32](#)

A hash code for the specified object.

Implements

[IEqualityComparer\(T\).GetHashCode\(T\)](#)

Exceptions

| Exception | Condition |
|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| ArgumentNullException | The type of <i>obj</i> is a reference type and <i>obj</i> is a null reference (Nothing in Visual Basic). |

See Also



[Vertex Class](#)

[GetHashCode Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetNumberOfEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
|  | GetNumberOfEdges(EdgeType, Direction) | Gets the number of edges from or to this vertex and for the given edge type. |
|  | GetNumberOfEdges(EdgeType, Vertex, Direction) | Gets the number of edges from or to this vertex for the given edge type and the given other Vertex. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.GetNumberOfEdges Method (EdgeType, Direction)

Gets the number of edges from or to this vertex and for the given edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long GetNumberOfEdges (
    EdgeType edgeType,
    Direction dir
)
```

VB

```
Public Function GetNumberOfEdges (
    edgeType As EdgeType,
    dir As Direction
) As Long
```

C++

```
public:
long long GetNumberOfEdges (
    EdgeType^ edgeType,
    Direction dir
)
```

F#

```
member GetNumberOfEdges :
    edgeType : EdgeType *
    dir : Direction -> int64
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

an EdgeType

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

direction, one of: Out, In, Both

Return Value

Type: [Int64](#)

The number of edges.

See Also

[Vertex Class](#)

VelocityDB Class Library

[GetNumberOfEdges Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetNumberOfEdges Method (EdgeType, Vertex, Direction)

Gets the number of edges from or to this vertex for the given edge type and the given other Vertex.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long GetNumberOfEdges (  
    EdgeType edgeType,  
    Vertex headVertex,  
    Direction dir  
)
```

VB

```
Public Function GetNumberOfEdges (  
    edgeType As EdgeType,  
    headVertex As Vertex,  
    dir As Direction  
) As Long
```

C++

```
public:  
long long GetNumberOfEdges (  
    EdgeType^ edgeType,  
    Vertex^ headVertex,  
    Direction dir  
)
```

F#

```
member GetNumberOfEdges :  
    edgeType : EdgeType *  
    headVertex : Vertex *  
    dir : Direction -> int64
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

an EdgeType

headVertex

Type: [VelocityGraph.Vertex](#)

Vertex at other end of the edge

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

direction, one of: Out, In, Both

VelocityDB Class Library

Return Value

Type: [Int64](#)

The number of edges.

See Also



[Vertex Class](#)

[GetNumberOfEdges Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetProperty Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetProperty(String) | Return the object value associated with the provided string key. If no value exists for that key, return null. (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetProperty(PropertyType) | Gets the Value for the given Property id |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.GetProperty Method (String)

Return the object value associated with the provided string key. If no value exists for that key, return null.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the key/value property

Return Value

Type: [Object](#)

the object value related to the string key

Implements

[IElement.GetProperty\(String\)](#)

[IElement.GetProperty\(String\)](#)

See Also

[Vertex Class](#)

VelocityDB Class Library

[GetProperty Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetProperty Method (PropertyType)

Gets the Value for the given Property id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetProperty(  
    PropertyType property  
)
```

VB

```
Public Function GetProperty (  
    property As PropertyType  
) As Object
```

C++

```
public:  
Object^ GetProperty(  
    PropertyType^ property  
)
```

F#

```
member GetProperty :  
    property : PropertyType -> Object
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

Property type identifier.

Return Value

Type: [Object](#)

the property value

See Also

[Vertex Class](#)

[GetProperty Overload](#)

[VelocityGraph Namespace](#)

Vertex.GetPropertyKeys Method

Return all the keys associated with the vertex.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

the set of all string keys associated with the vertex

Implements

[IElement.GetPropertyKeys\(\)](#)

[IElement.GetPropertyKeys\(\)](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.GetVertices Method

Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge).

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetVertices (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetVertices :
    direction : Direction *
    labels : string[] -> IEnumerable<IVertex>
override GetVertices :
    direction : Direction *
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

the direction of the edges of the adjacent vertices

labels

Type: [System.String\[\]](#)

the labels of the edges of the adjacent vertices

VelocityDB Class Library

Return Value

Type: [IEnumerable\(IVertex\)](#)

an IEnumerable of adjacent vertices

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Query Method

Uses [DefaultVertexQuery](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

Query interface

Implements

[IVertex.Query\(\)](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.RelatedVertices Method

[Missing <summary> documentation for "M:VelocityGraph.Vertex.RelatedVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.Collections.Generic.ISet{VelocityGraph.EdgeType})"]

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<Vertex> RelatedVertices (
    Direction dir = Direction.Both,
    ISet<EdgeType> edgeTypesToTraverse = null
)
```

VB

```
Public Function RelatedVertices (
    Optional dir As Direction = Direction.Both,
    Optional edgeTypesToTraverse As ISet(Of EdgeType) = Nothing
) As List(Of Vertex)
```

C++

```
public:
List<Vertex^>^ RelatedVertices(
    Direction dir = Direction::Both,
    ISet<EdgeType^>^ edgeTypesToTraverse = nullptr
)
```

F#

```
member RelatedVertices :
    ?dir : Direction *
    ?edgeTypesToTraverse : ISet<EdgeType>
(* Defaults:
    let_dir = defaultArg dir Direction.Both
    let_edgeTypesToTraverse = defaultArg edgeTypesToTraverse null
*)
-> List<Vertex>
```

Parameters

dir (Optional)

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="dir"/> documentation for "M:VelocityGraph.Vertex.RelatedVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.Collections.Generic.ISet{VelocityGraph.EdgeType})"]

edgeTypesToTraverse (Optional)

Type: [System.Collections.Generic.ISet\(EdgeType\)](#)

[Missing <param name="edgeTypesToTraverse"/> documentation for "M:VelocityGraph.Vertex.RelatedVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.Collections.Generic.ISet{VelocityGraph.EdgeType})"]

Return Value

Type: [List\(Vertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Vertex.RelatedVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.Collections.Generic.ISet{VelocityGraph.EdgeType})"]

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Remove Method

Removes this [Vertex](#) from the [Graph](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

[IElement.Remove\(\)](#)

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.RemoveProperty Method

Un-assigns a key/value property from the vertex. The object value of the removed property is returned.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the property to remove from the vertex

Return Value

Type: [Object](#)

the object value associated with that key prior to removal

Implements

[IElement.RemoveProperty\(String\)](#)

[IElement.RemoveProperty\(String\)](#)



See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.SetProperty Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | SetProperty(String, Object) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | SetProperty(PropertyType, IComparable) | Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten. |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.SetProperty Method (String, Object)

Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

the string key of the property

value

Type: [System.Object](#)

the object value of the property

VelocityDB Class Library

Implements

[IElement.SetProperty\(String, Object\)](#)

[IElement.SetProperty\(String, Object\)](#)

See Also

[Vertex Class](#)

[SetProperty Overload](#)

[VelocityGraph Namespace](#)

Vertex.SetProperty Method (PropertyType, IComparable)

Assign a key/value property to the vertex. If a value already exists for this key, then the previous key/value is overwritten.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetProperty(  
    PropertyType property,  
    IComparable v  
)
```

VB

```
Public Sub SetProperty (  
    property As PropertyType,  
    v As IComparable  
)
```

C++

```
public:  
void SetProperty(  
    PropertyType^ property,  
    IComparable^ v  
)
```

F#

```
member SetProperty :  
    property : PropertyType *  
    v : IComparable -> unit
```

Parameters

property

Type: [VelocityGraph.PropertyType](#)

The property type to set

v

Type: [System.IComparable](#)

the property value

See Also

[Vertex Class](#)

[SetProperty Overload](#)

[VelocityGraph Namespace](#)

Vertex.ToString Method

Returns a string that represents the current object.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A string that represents the current object.

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Traverse Method

Overload List

| | Name | Description |
|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| ⇒ | Traverse(Direction, ISet(EdgeType)) | Selects all neighbor Vertices from or to this vertex and for the given edge types. |
| ⇒ | Traverse(EdgeType, Direction) | Selects all neighbor Vertices from or to this vertex and for the given edge type. |
| ⇒ | Traverse(Int32, Boolean, Direction, Vertex, ISet(EdgeType), ISet(VertexType), ISet(VertexType), ISet(Vertex), ISet(Vertex), ISet(Edge), ISet(Edge), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), Func(Vertex, Boolean), Func(Edge, Boolean), Func(List(Edge), Boolean)) | Traverses graph from this Vertex to a target Vertex using Breadth-first search like in Dijkstra's algorithm |

See Also

[Vertex Class](#)

[VelocityGraph Namespace](#)

Vertex.Traverse Method (Direction, ISet(EdgeType))

Selects all neighbor Vertices from or to this vertex and for the given edge types.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Dictionary<Vertex, HashSet<Edge>> Traverse (  
    Direction dir,  
    ISet<EdgeType> edgeTypesToTraverse = null  
)
```

VB

```
Public Function Traverse (  
    dir As Direction,  
    Optional edgeTypesToTraverse As ISet(Of EdgeType) = Nothing  
) As Dictionary(Of Vertex, HashSet(Of Edge))
```

C++

```
public:  
Dictionary<Vertex^, HashSet<Edge^>>^ Traverse (  
    Direction dir,  
    ISet<EdgeType^>^ edgeTypesToTraverse = nullptr  
)
```

F#

```
member Traverse :  
    dir : Direction *  
    ?edgeTypesToTraverse : ISet<EdgeType>  
(* Defaults:  
    let _edgeTypesToTraverse = defaultArg edgeTypesToTraverse null  
)  
-> Dictionary<Vertex, HashSet<Edge>>
```

Parameters

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Direction to traverse edges

edgeTypesToTraverse (Optional)

Type: [System.Collections.Generic.ISet\(EdgeType\)](#)

the type of edges to follow, by default null which means follow all edge types

Return Value

Type: [Dictionary\(Vertex, HashSet\(Edge\)\)](#)

All paths to neighbor vertices

VelocityDB Class Library

See Also

[Vertex Class](#)

[Traverse Overload](#)

[VelocityGraph Namespace](#)

Vertex.Traverse Method (EdgeType, Direction)

Selects all neighbor Vertices from or to this vertex and for the given edge type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Dictionary<Vertex, HashSet<Edge>> Traverse (  
    EdgeType etype,  
    Direction dir  
)
```

VB

```
Public Function Traverse (  
    etype As EdgeType,  
    dir As Direction  
) As Dictionary(Of Vertex, HashSet(Of Edge))
```

C++

```
public:  
Dictionary<Vertex^, HashSet<Edge^>>^ Traverse (  
    EdgeType^ etype,  
    Direction dir  
)
```

F#

```
member Traverse :  
    etype : EdgeType *  
    dir : Direction -> Dictionary<Vertex, HashSet<Edge>>
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

Edge type identifier.

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Direction to traverse edges

Return Value

Type: [Dictionary\(Vertex, HashSet\(Edge\)\)](#)

Dictionary of vertex key with edge path(s) to vertex

See Also

[Vertex Class](#)

VelocityDB Class Library

[Traverse Overload](#)

[VelocityGraph Namespace](#)

Vertex.Traverse Method (Int32, Boolean, Direction, Vertex, ISet(EdgeType), ISet(VertexType), ISet(VertexType), ISet(Vertex), ISet(Vertex), ISet(Edge), ISet(Edge), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), ISet(PropertyType), Func(Vertex, Boolean), Func(Edge, Boolean), Func(List(Edge), Boolean))

Traverses graph from this Vertex to a target Vertex using Breadth-first search like in Dijkstra's algorithm

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<List<Edge>> Traverse (
    int maxHops,
    bool all = true,
    Direction dir = Direction.Both,
    Vertex toVertex = null,
    ISet<EdgeType> edgeTypesToTraverse = null,
    ISet<VertexType> includedVertexTypes = null,
    ISet<VertexType> excludedVertexTypes = null,
    ISet<Vertex> includedVertices = null,
    ISet<Vertex> excludedVertices = null,
    ISet<Edge> includedEdges = null,
    ISet<Edge> excludedEdges = null,
    ISet<PropertyType> includedVertexProperty = null,
    ISet<PropertyType> excludedVertexProperty = null,
    ISet<PropertyType> includedEdgeProperty = null,
    ISet<PropertyType> excludedEdgeProperty = null,
    Func<Vertex, bool> validateVertex = null,
    Func<Edge, bool> validateEdge = null,
    Func<List<Edge>, bool> validateEdges = null
)
```

VB

```
Public Function Traverse (
    maxHops As Integer,
    Optional all As Boolean = true,
    Optional dir As Direction = Direction.Both,
    Optional toVertex As Vertex = Nothing,
    Optional edgeTypesToTraverse As ISet(Of EdgeType) = Nothing,
    Optional includedVertexTypes As ISet(Of VertexType) = Nothing,
    Optional excludedVertexTypes As ISet(Of VertexType) = Nothing,
    Optional includedVertices As ISet(Of Vertex) = Nothing,
    Optional excludedVertices As ISet(Of Vertex) = Nothing,
    Optional includedEdges As ISet(Of Edge) = Nothing,
    Optional excludedEdges As ISet(Of Edge) = Nothing,
    Optional includedVertexProperty As ISet(Of PropertyType) = Nothing,
    Optional excludedVertexProperty As ISet(Of PropertyType) = Nothing,
    Optional includedEdgeProperty As ISet(Of PropertyType) = Nothing,
```

```

Optional excludedEdgeProperty As ISet(Of PropertyType) = Nothing,
Optional validateVertex As Func(Of Vertex, Boolean) = Nothing,
Optional validateEdge As Func(Of Edge, Boolean) = Nothing,
Optional validateEdges As Func(Of List(Of Edge), Boolean) = Nothing
) As List(Of List(Of Edge))

```

C++

```

public:
List<List<Edge^>>^ Traverse(
    int maxHops,
    bool all = true,
    Direction dir = Direction::Both,
    Vertex^ toVertex = nullptr,
    ISet<EdgeType^>^ edgeTypesToTraverse = nullptr,
    ISet<VertexType^>^ includedVertexTypes = nullptr,
    ISet<VertexType^>^ excludedVertexTypes = nullptr,
    ISet<Vertex^>^ includedVertices = nullptr,
    ISet<Vertex^>^ excludedVertices = nullptr,
    ISet<Edge^>^ includedEdges = nullptr,
    ISet<Edge^>^ excludedEdges = nullptr,
    ISet<PropertyType^>^ includedVertexProperty = nullptr,
    ISet<PropertyType^>^ excludedVertexProperty = nullptr,
    ISet<PropertyType^>^ includedEdgeProperty = nullptr,
    ISet<PropertyType^>^ excludedEdgeProperty = nullptr,
    Func<Vertex^, bool>^ validateVertex = nullptr,
    Func<Edge^, bool>^ validateEdge = nullptr,
    Func<List<Edge^>, bool>^ validateEdges = nullptr
)

```

F#

```

member Traverse :
    maxHops : int *
    ?all : bool *
    ?dir : Direction *
    ?toVertex : Vertex *
    ?edgeTypesToTraverse : ISet<EdgeType> *
    ?includedVertexTypes : ISet<VertexType> *
    ?excludedVertexTypes : ISet<VertexType> *
    ?includedVertices : ISet<Vertex> *
    ?excludedVertices : ISet<Vertex> *
    ?includedEdges : ISet<Edge> *
    ?excludedEdges : ISet<Edge> *
    ?includedVertexProperty : ISet<PropertyType> *
    ?excludedVertexProperty : ISet<PropertyType> *
    ?includedEdgeProperty : ISet<PropertyType> *
    ?excludedEdgeProperty : ISet<PropertyType> *
    ?validateVertex : Func<Vertex, bool> *
    ?validateEdge : Func<Edge, bool> *
    ?validateEdges : Func<List<Edge>, bool>
(* Defaults:
    let_all = defaultArg all true
    let_dir = defaultArg dir Direction.Both
    let_toVertex = defaultArg toVertex null
    let_edgeTypesToTraverse = defaultArg edgeTypesToTraverse null

```

```
let_includedVertexTypes = defaultArg includedVertexTypes null
let_excludedVertexTypes = defaultArg excludedVertexTypes null
let_includedVertices = defaultArg includedVertices null
let_excludedVertices = defaultArg excludedVertices null
let_includedEdges = defaultArg includedEdges null
let_excludedEdges = defaultArg excludedEdges null
let_includedVertexProperty = defaultArg includedVertexProperty null
let_excludedVertexProperty = defaultArg excludedVertexProperty null
let_includedEdgeProperty = defaultArg includedEdgeProperty null
let_excludedEdgeProperty = defaultArg excludedEdgeProperty null
let_validateVertex = defaultArg validateVertex null
let_validateEdge = defaultArg validateEdge null
let_validateEdges = defaultArg validateEdges null
*)
-> List<List<Edge>>
```

Parameters

maxHops

Type: [System.Int32](#)

maximum number of hops from this Vertex

all (Optional)

Type: [System.Boolean](#)

find or not find all paths to goal Vertex

dir (Optional)

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Direction to traverse edges

toVertex (Optional)

Type: [VelocityGraph.Vertex](#)

the goal Vertex. If null, finds all paths

edgeTypesToTraverse (Optional)

Type: [System.Collections.Generic.ISet\(EdgeType\)](#)

the type of edges to follow, by default null which means follow all edge types

includedVertexTypes (Optional)

Type: [System.Collections.Generic.ISet\(VertexType\)](#)

the type of vertices's to follow, by default null which means follow all vertex types

excludedVertexTypes (Optional)

Type: [System.Collections.Generic.ISet\(VertexType\)](#)

the type of vertices's not to follow, by default null

includedVertices (Optional)

Type: [System.Collections.Generic.ISet\(Vertex\)](#)

one or more Vertex instances that MUST be in the path for the path to be traversed i.e. if a path does exist to the specified toVertex, but does not include all the instances in includedVertices set, the Traverse method will exclude that path

excludedVertices (Optional)

Type: [System.Collections.Generic.ISet\(Vertex\)](#)

one or more Vertex instances that MUST NOT be in the path for the path to be traversed i.e. if a path does exist to the specified toVertex, but does include any of the instances in includedVertices set, the Traverse method will exclude that path

includedEdges (Optional)

Type: [System.Collections.Generic.ISet\(Edge\)](#)

one or more Edge instances that MUST be in the path for the path to be traversed i.e. if a path does exist to the specified toVertex, but does not include all the instances in includedEdges set, the Traverse method will exclude that path

excludedEdges (Optional)

Type: [System.Collections.Generic.ISet\(Edge\)](#)

one or more Edge instances that MUST NOT be in the path for the path to be traversed i.e. if a path does exist to the specified toVertex, but does include any of the instances in includedEdges set, the Traverse method will exclude that path

includedVertexProperty (Optional)

Type: [System.Collections.Generic.ISet\(PropertyType\)](#)

One or more Vertex property types that MUST be in the path for the path to be accepted i.e. if a path does exist to the specified toVertex, but does not include all of the Vertex properties in the set, the Traverse method will exclude that path

excludedVertexProperty (Optional)

Type: [System.Collections.Generic.ISet\(PropertyType\)](#)

One or more Vertex property types that MUST NOT be in the path for the path to be accepted i.e. if a path does exist to the specified toVertex, but does include any of the Vertex properties in the set, the Traverse method will exclude that path

includedEdgeProperty (Optional)

Type: [System.Collections.Generic.ISet\(PropertyType\)](#)

One or more Vertex property types that MUST be in the path for the path to be accepted i.e. if a path does exist to the specified toVertex, but does not include all of the Vertex properties in the set, the Traverse method will exclude that path

excludedEdgeProperty (Optional)

Type: [System.Collections.Generic.ISet\(PropertyType\)](#)

One or more Edge property types that MUST NOT be in the path for the path to be accepted i.e. if a path does exist to the specified toVertex, but does include any of the Edge properties in the set, the Traverse method will exclude that path

validateVertex (Optional)

Type: [System.Func\(Vertex, Boolean\)](#)

A function that will be called before accepting a Vertex in path to toVertex. If function returns true then this vertex is accepted in path; otherwise vertex is rejected

validateEdge (Optional)

Type: [System.Func\(Edge, Boolean\)](#)

A function that will be called before accepting an Edge in path to toVertex. If function returns true then this Edge is accepted in path; otherwise edge is rejected

validateEdges (Optional)

Type: [System.Func\(List\(Edge\), Boolean\)](#)

A function that will be called before accepting a candidate Edges list in path to toVertex. If function returns true then this Edge list is accepted in path; otherwise edge list is rejected

Return Value

Type: [List\(List\(Edge\)\)](#)

List of paths to goal Vertex

See Also

[Vertex Class](#)

[Traverse Overload](#)

[VelocityGraph Namespace](#)

VertexType Class

All vertices have a type that is identified as a `VertexType`. Each `VertexType` have a name, can be persisted and tracks all vertices of its type. The vertex type also knows about properties used by its type.

Inheritance Hierarchy

[System.Object](#)

[VelocityDb.OptimizedPersistable](#)

VelocityGraph.VertexType

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class VertexType : OptimizedPersistable
```

VB

```
<SerializableAttribute>
Public Class VertexType
    Inherits OptimizedPersistable
```

C++






```
[SerializableAttribute]
public ref class VertexType : public OptimizedPersistable
```

F#

```
[<SerializableAttribute>]
type VertexType =
    class
        inherit OptimizedPersistable
    end
```

The **VertexType** type exposes the following members.

Properties



| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------|
|  | EdgeTypes | All edge types connected with this vertex type. |
|  | MyGraph | Graph for which this VertexType belongs to |
|  | SubTypes | Sub types of this VertexType |
|  | TypeId | Get the unique id of the vertex type. |
|  | TypeName | Gets the name of this VertexType |

Methods

| Name | Description |
|---------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CountVertices | Count of vertices |
| FindProperty | Gets the associated PropertyType given a property type name or null if such property type doesn't exist. |
| GetEdges(EdgeType, Direction) | Enumerates edges connected with this vertex type |
| GetEdges(Vertex, Direction) | Get all edges found from a given Vertex |
| GetEdges(EdgeType, Vertex, Direction, Vertex) | Get all edges found between two vertices |
| GetNumberOfEdges(EdgeType, Direction) | Get the number of edges of a certain type that can be found and an edge direction |
| GetNumberOfEdges(EdgeType, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id and an edge direction |
| GetNumberOfEdges(EdgeType, Int32, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id, another vertex id at other end and an edge direction |
| GetPropertyKeys | Return all the keys associated with the vertex type. |
| GetPropertyTypes | Return all the property types associated with vertex type. |
| GetPropertyValue | Get the property value for a Vertex |
| GetTopNumberOfEdges | Get the top vertices with the most number of edges of the given edge type |
| GetVertex | Instantiates a Vertex if it exist |
| GetVertexIds | Get existing Vertex ids for this type |
| GetVertices(Boolean) | Enumerates all vertices for the given type |
| GetVertices(EdgeType, Vertex, Direction) | Get an enumeration of existing vertices of this type found by following an edge type |
| NewProperty | Creates a new Property. |
| NewVertex | Creates a new Vertex |
| Remove | Removes this VertexType from a graph. An exception is thrown if the VertexType is in use. |
| RemoveVertex | Removes a vertex from this type and graph |
| SetPropertyValue | Sets a property value |
| ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |
| Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override |

| | | |
|--|--|----------------------------------------------------------------------------------------------------------------------|
| | | this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase).) |
|--|--|----------------------------------------------------------------------------------------------------------------------|

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities.) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities.) |






See Also

[VelocityGraph Namespace](#)

VertexType.VertexType Properties

The [VertexType](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------------|
|  | EdgeTypes | All edge types connected with this vertex type. |
|  | MyGraph | Graph for which this VertexType belongs to |
|  | SubTypes | Sub types of this VertexType |
|  | TypeId | Get the unique id of the vertex type. |
|  | TypeName | Gets the name of this VertexType |

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.EdgeTypes Property

All edge types connected with this vertex type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BTreeSet<EdgeType> EdgeTypes { get; }
```

VB

```
Public ReadOnly Property EdgeTypes As BTreeSet(Of EdgeType)  
    Get
```

C++

```
public:  
property BTreeSet<EdgeType^>^ EdgeTypes {  
    BTreeSet<EdgeType^>^ get ();  
}
```

F#

```
member EdgeTypes : BTreeSet<EdgeType> with get
```

Property Value

Type: [BTreeSet\(EdgeType\)](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.MyGraph Property

[Graph](#)

for which this [VertexType](#) belongs to

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Graph MyGraph { get; }
```

VB

```
Public ReadOnly Property MyGraph As Graph  
    Get
```

C++

```
public:  
property Graph^ MyGraph {  
    Graph^ get ();  
}
```

F#

```
member MyGraph : Graph with get
```

Property Value

Type: [Graph](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.SubTypes Property

Sub types of this [VertexType](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<VertexType> SubTypes { get; }
```

VB

```
Public ReadOnly Property SubTypes As List(Of VertexType)  
    Get
```

C++

```
public:  
property List<VertexType^>^ SubTypes {  
    List<VertexType^>^ get ();  
}
```

F#

```
member SubTypes : List<VertexType> with get
```

Property Value

Type: [List\(VertexType\)](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.TypeId Property

Get the unique id of the vertex type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int TypeId { get; }
```

VB

```
Public ReadOnly Property TypeId As Integer  
    Get
```

C++

```
public:  
property int TypeId {  
    int get ();  
}
```

F#

```
member TypeId : int with get
```

Property Value

Type: [Int32](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.TypeName Property

Gets the name of this [VertexType](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TypeName { get; }
```

VB

```
Public ReadOnly Property TypeName As String  
    Get
```

C++

```
public:  
property String^ TypeName {  
    String^ get ();  
}
```

F#

```
member TypeName : string with get
```

Property Value

Type: [String](#)

See Also

[VertexType Class](#)


[VelocityGraph Namespace](#)

VertexType.VertexType Methods



The [VertexType](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| CountVertices | Count of vertices |
| FindProperty | Gets the associated PropertyType given a property type name or null if such property type doesn't exist. |
| GetEdges(EdgeType, Direction) | Enumerates edges connected with this vertex type |
| GetEdges(Vertex, Direction) | Get all edges found from a given Vertex |
| GetEdges(EdgeType, Vertex, Direction, Vertex) | Get all edges found between two vertices |
| GetNumberOfEdges(EdgeType, Direction) | Get the number of edges of a certain type that can be found and an edge direction |
| GetNumberOfEdges(EdgeType, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id and an edge direction |
| GetNumberOfEdges(EdgeType, Int32, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id, another vertex id at other end and an edge direction |
| GetPropertyKeys | Return all the keys associated with the vertex type. |
| GetPropertyTypes | Return all the property types associated with vertex type. |
| GetPropertyValue | Get the property value for a Vertex |
| GetTopNumberOfEdges | Get the top vertices with the most number of edges of the given edge type |
| GetVertex | Instantiates a Vertex if it exist |
| GetVertexIds | Get existing Vertex ids for this type |
| GetVertices(Boolean) | Enumerates all vertices for the given type |
| GetVertices(EdgeType, Vertex, Direction) | Get an enumeration of existing vertices of this type found by following an edge type |
| NewProperty | Creates a new Property. |
| NewVertex | Creates a new Vertex |
| Remove | Removes this VertexType from a graph. An exception is thrown if the VertexType is in use. |
| RemoveVertex | Removes a vertex from this type and graph |
| SetPropertyValue | Sets a property value |
| ToString | Displays class name plus object id (Overrides OptimizedPersistable.ToString() .) |

| | | |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Unpersist | Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes. (Overrides OptimizedPersistable.Unpersist(SessionBase) .) |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------|
|  | ToStringDetails(SessionBase, Boolean) | Overloaded. Object details as a string (Defined by Utilities .) |
|  | ToStringDetails(Schema, TypeVersion, Boolean) | Overloaded. Currently only used by Database Manager (Defined by Utilities .) |

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.CountVertices Method

Count of verticies

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long CountVertices ()
```

VB

```
Public Function CountVertices As Long
```

C++

```
public:  
long long CountVertices ()
```

F#

```
member CountVertices : unit -> int64
```

Return Value

Type: [Int64](#)

the number of verices that exist for this type in this [Graph](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.FindProperty Method

Gets the associated [PropertyType](#) given a property type name or null if such property type doesn't exist.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyType FindProperty(  
    string name,  
    bool doBaseType = true  
)
```

VB

```
Public Function FindProperty (  
    name As String,  
    Optional doBaseType As Boolean = true  
) As PropertyType
```

C++

```
public:  
PropertyType^ FindProperty(  
    String^ name,  
    bool doBaseType = true  
)
```

F#

```
member FindProperty :  
    name : string *  
    ?doBaseType : bool  
(* Defaults:  
    let _doBaseType = defaultArg doBaseType true  
)  
-> PropertyType
```

Parameters

name

Type: [System.String](#)

A property type name

doBaseType (Optional)

Type: [System.Boolean](#)

[Missing <param name="doBaseType"/> documentation for "M:VelocityGraph.VertexType.FindProperty(System.String,System.Boolean)"]

VelocityDB Class Library

Return Value

Type: [PropertyType](#)

The property type or null




See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|---------------------------------------------------------|
|  | GetEdges(EdgeType, Direction) | Enumerates edges connected with this vertex type |
|  | GetEdges(Vertex, Direction) | Get all edges found from a given Vertex |
|  | GetEdges(EdgeType, Vertex, Direction, Vertex) | Get all edges found between two vertices |

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetEdges Method (EdgeType, Direction)

Enumerates edges connected with this vertex type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    EdgeType etype,  
    Direction dir  
)
```

VB

```
Public Function GetEdges (  
    etype As EdgeType,  
    dir As Direction  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
IEnumerable<IEdge^>^ GetEdges (  
    EdgeType^ etype,  
    Direction dir  
)
```

F#

```
member GetEdges :  
    etype : EdgeType *  
    dir : Direction -> IEnumerable<IEdge>
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

A type of edge type to look for

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Direction of edge

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.VertexType.GetEdges(VelocityGraph.EdgeType,VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

See Also

[VertexType Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetEdges Method (Vertex, Direction)

Get all edges found from a given [Vertex](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    Vertex vertex1,  
    Direction dir  
)
```

VB

```
Public Function GetEdges (  
    vertex1 As Vertex,  
    dir As Direction  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
IEnumerable<IEdge^>^ GetEdges (  
    Vertex^ vertex1,  
    Direction dir  
)
```

F#

```
member GetEdges :  
    vertex1 : Vertex *  
    dir : Direction -> IEnumerable<IEdge>
```

Parameters

vertex1

Type: [VelocityGraph.Vertex](#)

A [Vertex](#) id

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

follow edges in this direction

Return Value

Type: [IEnumerable\(IEdge\)](#)

an enumeration of edges

See Also

[VertexType Class](#)

VelocityDB Class Library

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetEdges Method (EdgeType, Vertex, Direction, Vertex)

Get all edges found between two vertices

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    EdgeType edgeType,
    Vertex vertex1,
    Direction dir,
    Vertex vertex2 = null
)
```

VB

```
Public Function GetEdges (
    edgeType As EdgeType,
    vertex1 As Vertex,
    dir As Direction,
    Optional vertex2 As Vertex = Nothing
) As IEnumerable(Of IEdge)
```

C++

```
public:
IEnumerable<IEdge^>^ GetEdges (
    EdgeType^ edgeType,
    Vertex^ vertex1,
    Direction dir,
    Vertex^ vertex2 = nullptr
)
```

F#

```
member GetEdges :
    edgeType : EdgeType *
    vertex1 : Vertex *
    dir : Direction *
    ?vertex2 : Vertex
(* Defaults:
    let _vertex2 = defaultArg vertex2 null
*)
-> IEnumerable<IEdge>
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

Restrict to this type of edge

VelocityDB Class Library

vertex1

Type: [VelocityGraph.Vertex](#)

Start [Vertex](#)

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Follow edges in this direction

vertex2 (Optional)

Type: [VelocityGraph.Vertex](#)

End [Vertex](#)

Return Value

Type: [IEnumerable\(IEdge\)](#)

An enumeration of edges

See Also




[VertexType Class](#)

[GetEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetNumberOfEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetNumberOfEdges(EdgeType, Direction) | Get the number of edges of a certain type that can be found and an edge direction |
|  | GetNumberOfEdges(EdgeType, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id and an edge direction |
|  | GetNumberOfEdges(EdgeType, Int32, Int32, Direction) | Get the number of edges of a certain type that can be found associated with a vertex id, another vertex id at other end and an edge direction |

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetNumberOfEdges Method (EdgeType, Direction)

Get the number of edges of a certain type that can be found and an edge direction

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long GetNumberOfEdges (  
    EdgeType etype,  
    Direction dir  
)
```

VB

```
Public Function GetNumberOfEdges (  
    etype As EdgeType,  
    dir As Direction  
) As Long
```

C++

```
public:  
long long GetNumberOfEdges (  
    EdgeType^ etype,  
    Direction dir  
)
```

F#

```
member GetNumberOfEdges :  
    etype : EdgeType *  
    dir : Direction -> int64
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

Use this type of edge

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Edge direction to follow

Return Value

Type: [Int64](#)

Number of edges found

See Also

[VertexType Class](#)

VelocityDB Class Library

[GetNumberOfEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetNumberOfEdges Method (EdgeType, Int32, Direction)

Get the number of edges of a certain type that can be found associated with a vertex id and an edge direction

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long GetNumberOfEdges (
    EdgeType etype,
    int vertexId,
    Direction dir
)
```

VB

```
Public Function GetNumberOfEdges (
    etype As EdgeType,
    vertexId As Integer,
    dir As Direction
) As Long
```

C++

```
public:
long long GetNumberOfEdges (
    EdgeType^ etype,
    int vertexId,
    Direction dir
)
```

F#

```
member GetNumberOfEdges :
    etype : EdgeType *
    vertexId : int *
    dir : Direction -> int64
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

Type of edges to look for

vertexId

Type: [System.Int32](#)

Id of a [Vertex](#)

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

VelocityDB Class Library

Edge direction to use

Return Value

Type: [Int64](#)

The number of edges found

See Also

[VertexType Class](#)

[GetNumberOfEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetNumberOfEdges Method (EdgeType, Int32, Int32, Direction)

Get the number of edges of a certain type that can be found associated with a vertex id, another vertex id at other end and an edge direction

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long GetNumberOfEdges (
    EdgeType edgeType,
    int vertexId,
    int vertexId2,
    Direction dir
)
```

VB

```
Public Function GetNumberOfEdges (
    edgeType As EdgeType,
    vertexId As Integer,
    vertexId2 As Integer,
    dir As Direction
) As Long
```

C++

```
public:
long long GetNumberOfEdges (
    EdgeType^ edgeType,
    int vertexId,
    int vertexId2,
    Direction dir
)
```

F#

```
member GetNumberOfEdges :
    edgeType : EdgeType *
    vertexId : int *
    vertexId2 : int *
    dir : Direction -> int64
```

Parameters

edgeType

Type: [VelocityGraph.EdgeType](#)

[Missing <param name="edgeType"/> documentation for "M:VelocityGraph.VertexType.GetNumberOfEdges(VelocityGraph.EdgeType,System.Int32,System.Int32,VelocityGraph.Frontenac.Blueprints.Direction)"]

vertexId

Type: [System.Int32](#)

Id of a [Vertex](#)

vertexId2

Type: [System.Int32](#)

Id of a [Vertex](#)

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Edge direction to use

Return Value

Type: [Int64](#)

The number of edges found

See Also

[VertexType Class](#)

[GetNumberOfEdges Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetPropertyKeys Method

Return all the keys associated with the vertex type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Function GetPropertyKeys As IEnumerable (Of String)
```

C++

```
public:  
IEnumerable<String^>^ GetPropertyKeys ()
```

F#

```
member GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

the set of all string keys associated with the vertex type

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetPropertyTypes Method

Return all the property types associated with vertex type.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<PropertyType> GetPropertyTypes ()
```

VB

```
Public Function GetPropertyTypes As IEnumerable (Of PropertyType)
```

C++

```
public:  
IEnumerable<PropertyType^> GetPropertyTypes ()
```

F#

```
member GetPropertyTypes : unit -> IEnumerable<PropertyType>
```

Return Value

Type: [IEnumerable\(PropertyType\)](#)

the set of property types associated with the vertex type

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetPropertyValue Method

Get the property value for a [Vertex](#)

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetPropertyValue(  
    int vertexId,  
    PropertyType propertyType  
)
```

VB

```
Public Function GetPropertyValue (  
    vertexId As Integer,  
    propertyType As PropertyType  
) As Object
```

C++

```
public:  
Object^ GetPropertyValue (  
    int vertexId,  
    PropertyType^ propertyType  
)
```

F#

```
member GetPropertyValue :  
    vertexId : int *  
    propertyType : PropertyType -> Object
```

Parameters

vertexId

Type: [System.Int32](#)

Id of [Vertex](#)

propertyType

Type: [VelocityGraph.PropertyType](#)

Type of property

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.VertexType.GetPropertyValue(System.Int32,VelocityGraph.PropertyType)"]

VelocityDB Class Library

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetTopNumberOfEdges Method

Get the top vertices with the most number of edges of the given edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex[] GetTopNumberOfEdges (  
    EdgeType etype,  
    int howMany,  
    Direction dir  
)
```

VB

```
Public Function GetTopNumberOfEdges (  
    etype As EdgeType,  
    howMany As Integer,  
    dir As Direction  
) As Vertex()
```

C++

```
public:  
array<Vertex^>^ GetTopNumberOfEdges (  
    EdgeType^ etype,  
    int howMany,  
    Direction dir  
)
```

F#

```
member GetTopNumberOfEdges :  
    etype : EdgeType *  
    howMany : int *  
    dir : Direction -> Vertex[]
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

The edge type to look for

howMany

Type: [System.Int32](#)

How many top ones to collect

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

What end of edges to look at

VelocityDB Class Library

Return Value

Type: [Vertex](#)[]

Array of Vertices with the most edges of the given edge type

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetVertex Method

Instantiates a Vertex if it exist

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex GetVertex(  
    int vertexId,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

VB

```
Public Function GetVertex (  
    vertexId As Integer,  
    Optional polymorphic As Boolean = false,  
    Optional errorIfNotFound As Boolean = true  
) As Vertex
```

C++

```
public:  
Vertex^ GetVertex(  
    int vertexId,  
    bool polymorphic = false,  
    bool errorIfNotFound = true  
)
```

F#

```
member GetVertex :  
    vertexId : int *  
    ?polymorphic : bool *  
    ?errorIfNotFound : bool  
(* Defaults:  
    let _polymorphic = defaultArg polymorphic false  
    let _errorIfNotFound = defaultArg errorIfNotFound true  
)  
-> Vertex
```

Parameters

vertexId

Type: [System.Int32](#)

id of Vertex we are looking for

polymorphic (Optional)

Type: [System.Boolean](#)

VelocityDB Class Library

If true and id isn't found in this VertexType continue search into sub types

errorIfNotFound (Optional)

Type: [System.Boolean](#)

Indicate what to do if Vertex does not exist

Return Value

Type: [Vertex](#)

A Vertex or null

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetVerticeIds Method

Get existing [Vertex](#) ids for this type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<int> GetVerticeIds ()
```

VB

```
Public Function GetVerticeIds As IEnumerable (Of Integer)
```

C++

```
public:  
IEnumerable<int>^ GetVerticeIds ()
```

F#

```
member GetVerticeIds : unit -> IEnumerable<int>
```

Return Value

Type: [IEnumerable\(Int32\)](#)

An enumeration of vertex ids



See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------|--------------------------------------------------------------------------------------|
|  | GetVertices(Boolean) | Enumerates all vertices for the given type |
|  | GetVertices(EdgeType, Vertex, Direction) | Get an enumeration of existing vertices of this type found by following an edge type |

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.GetVertices Method (Boolean)

Enumerates all vertices for the given type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Vertex> GetVertices(  
    bool polymorphic = false  
)
```

VB

```
Public Function GetVertices (  
    Optional polymorphic As Boolean = false  
) As IEnumerable(Of Vertex)
```

C++

```
public:  
IEnumerable<Vertex^> GetVertices(  
    bool polymorphic = false  
)
```

F#

```
member GetVertices :  
    ?polymorphic : bool  
(* Defaults:  
    let_polymorphic = defaultArg polymorphic false  
*)  
-> IEnumerable<Vertex>
```

Parameters

polymorphic (Optional)

Type: [System.Boolean](#)

If true, also include all vertices of sub types of this VertexType

Return Value

Type: [IEnumerable\(Vertex\)](#)

Enumeration of vertices

See Also

[VertexType Class](#)

[GetVertices Overload](#)

[VelocityGraph Namespace](#)

VertexType.GetVertices Method (EdgeType, Vertex, Direction)

Get an enumeration of existing vertices of this type found by following an edge type

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    EdgeType etype,  
    Vertex vertex1,  
    Direction dir  
)
```

VB

```
Public Function GetVertices (  
    etype As EdgeType,  
    vertex1 As Vertex,  
    dir As Direction  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
IEnumerable<IVertex^> GetVertices (  
    EdgeType^ etype,  
    Vertex^ vertex1,  
    Direction dir  
)
```

F#

```
member GetVertices :  
    etype : EdgeType *  
    vertex1 : Vertex *  
    dir : Direction -> IEnumerable<IVertex>
```

Parameters

etype

Type: [VelocityGraph.EdgeType](#)

Edge type to follow

vertex1

Type: [VelocityGraph.Vertex](#)

Vertex to start from

dir

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

Edge direction to follow

VelocityDB Class Library

Return Value

Type: [IEnumerable\(IVertex\)](#)
an enumeration of vertices

See Also

[VertexType Class](#)

[GetVertices Overload](#)

[VelocityGraph Namespace](#)

VertexType.NewProperty Method

Creates a new Property.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public PropertyType NewProperty(  
    string name,  
    DataType dt,  
    PropertyKind kind  
)
```

```
VB  
Public Function NewProperty (  
    name As String,  
    dt As DataType,  
    kind As PropertyKind  
) As PropertyType
```

```
C++  
public:  
PropertyType^ NewProperty(  
    String^ name,  
    DataType dt,  
    PropertyKind kind  
)
```

```
F#  
member NewProperty :  
    name : string *  
    dt : DataType *  
    kind : PropertyKind -> PropertyType
```

Parameters

name

Type: [System.String](#)

Unique name for the new Property.

dt

Type: [VelocityGraph.DataType](#)

Data type for the new Property.

kind

Type: [VelocityGraph.PropertyKind](#)

Property kind.

VelocityDB Class Library

Return Value

Type: [PropertyType](#)

Unique Property identifier.

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.NewVertex Method

Creates a new Vertex

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Vertex NewVertex(  
    int vId = 0  
)
```

VB

```
Public Function NewVertex (  
    Optional vId As Integer = 0  
) As Vertex
```

C++

```
public:  
Vertex^ NewVertex(  
    int vId = 0  
)
```

F#

```
member NewVertex :  
    ?vId : int  
(* Defaults:  
    let _vId = defaultArg vId 0  
*)  
-> Vertex
```

Parameters

vId (Optional)

Type: [System.Int32](#)

Optionally provide the Vertex id to use.

Return Value

Type: [Vertex](#)

The newly created [Vertex](#)

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.Remove Method

Removes this [VertexType](#) from a graph. An exception is thrown if the [VertexType](#) is in use.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove ()
```

VB

```
Public Sub Remove
```

C++

```
public:  
void Remove ()
```

F#

```
member Remove : unit -> unit
```

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.RemoveVertex Method

Removes a vertex from this type and graph

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    Vertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As Vertex  
)
```

C++

```
public:  
void RemoveVertex(  
    Vertex^ vertex  
)
```

F#

```
member RemoveVertex :  
    vertex : Vertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Vertex](#)

The vertex to remove

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.SetPropertyValue Method

Sets a property value

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetPropertyValue(  
    VertexType vt,  
    int vertexId,  
    PropertyType propertyType,  
    IComparable v  
)
```

VB

```
Public Sub SetPropertyValue (  
    vt As VertexType,  
    vertexId As Integer,  
    propertyType As PropertyType,  
    v As IComparable  
)
```

C++

```
public:  
void SetPropertyValue(  
    VertexType^ vt,  
    int vertexId,  
    PropertyType^ propertyType,  
    IComparable^ v  
)
```

F#

```
member SetPropertyValue :  
    vt : VertexType *  
    vertexId : int *  
    propertyType : PropertyType *  
    v : IComparable -> unit
```

Parameters

vt

Type: [VelocityGraph.VertexType](#)

[Missing <param name="vt"/> documentation for

"M:VelocityGraph.VertexType.SetPropertyValue(VelocityGraph.VertexType,System.Int32,VelocityGraph.PropertyType,System.IComparable)"]

vertexId

VelocityDB Class Library

Type: [System.Int32](#)

Id of [Vertex](#) for which to set property value

propertyType

Type: [VelocityGraph.PropertyType](#)

The type of property to set

v

Type: [System.IComparable](#)

the value to set the property to

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.ToString Method

Displays class name plus object id

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

A [String](#) containing class name and object id.

See Also

[VertexType Class](#)

[VelocityGraph Namespace](#)

VertexType.Unpersist Method

Removes an object from the persistent store and makes the object a transient object. It does not automatically make referenced objects unpersisted. Best way to do so is to override this virtual function in your own classes.

Namespace: [VelocityGraph](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Unpersist(  
    SessionBase session  
)
```

VB

```
Public Overrides Sub Unpersist (  
    session As SessionBase  
)
```

C++

```
public:  
virtual void Unpersist(  
    SessionBase^ session  
) override
```

F#

```
abstract Unpersist :  
    session : SessionBase -> unit  
override Unpersist :  
    session : SessionBase -> unit
```

Parameters

session

Type: [VelocityDb.Session.SessionBase](#)

The managing session

Implements

[IOptimizedPersistable.Unpersist\(SessionBase\)](#)

See Also







[VertexType Class](#)

[VelocityGraph Namespace](#)

VelocityGraph.Exceptions Namespace

The **VelocityGraph.Exceptions** namespace contains exception classes for the VelocityGraph Graph Database.

Classes

| Class | Description |
|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|  EdgeDoesNotExistException | Exception thrown when a edge id is specified for an edge type and the edge does not exist. |
|  EdgeTypeDoesNotExistException | Exception thrown when a edge type id is specified and the edge type does not exist. |
|  EdgeTypeInUseException | Exception thrown when at least one edge exist for a edge type and attempting to remove the edge type. |
|  InvalidHeadVertexTypeException | Exception thrown when the type of the head vertex doesn't match the required head VertexType of an EdgeType. |
|  InvalidPropertyIdException | Exception thrown when a type id is specified for a property that does not exist. |
|  InvalidTailVertexTypeException | Exception thrown when the type of the tail vertex doesn't match the required tail VertexType of an EdgeType. |
|  InvalidTypeIdException | Exception thrown when a type id is specified for a type that does not exist. |
|  PropertyTypeInUseException | Exception thrown when at least one vertex or edge uses a property type when attempting to remove the property type |
|  VertexAlreadyExistException | Exception thrown when a vertex id is specified for a vertex type and the vertex already exist. |
|  VertexDoesNotExistException | Exception thrown when a vertex id is specified for a vertex type and the vertex does not exist. |
|  VertexTypeDoesNotExistException | Exception thrown when a vertex type id is specified and the vertex type does not exist. |
|  VertexTypeInUseException | Exception thrown when at least one vertex exist for a vertex type and attempting to remove the vertex type. |

EdgeDoesNotExistException Class

Exception thrown when an edge id is specified for an edge type and the edge does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.EdgeDoesNotExistException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class EdgeDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class EdgeDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class EdgeDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type EdgeDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

EdgeTypeDoesNotExistException Class

Exception thrown when a edge type id is specified and the edge type does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.EdgeTypeDoesNotExistException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class EdgeTypeDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class EdgeTypeDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class EdgeTypeDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type EdgeTypeDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

EdgeTypeInUseException Class

Exception thrown when at least one edge exist for a edge type and attempting to remove the edge type.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.EdgeTypeInUseException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class EdgeTypeInUseException : Exception
```

VB

```
<SerializableAttribute>  
Public Class EdgeTypeInUseException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class EdgeTypeInUseException : public Exception
```

F#

```
[<SerializableAttribute>]  
type EdgeTypeInUseException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

InvalidHeadVertexTypeException Class

Exception thrown when the type of the head vertex doesn't match the required head VertexType of an EdgeType.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.InvalidHeadVertexTypeException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidHeadVertexTypeException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidHeadVertexTypeException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidHeadVertexTypeException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidHeadVertexTypeException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

InvalidPropertyIdException Class

Exception thrown when a type id is specified for a property that does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.InvalidPropertyIdException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidPropertyIdException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidPropertyIdException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidPropertyIdException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidPropertyIdException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

InvalidTailVertexTypeException Class

Exception thrown when the type of the tail vertex doesn't match the required tail VertexType of an EdgeType.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.InvalidTailVertexTypeException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidTailVertexTypeException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidTailVertexTypeException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidTailVertexTypeException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidTailVertexTypeException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

InvalidTypeIdException Class

Exception thrown when a type id is specified for a type that does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.InvalidTypeIdException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class InvalidTypeIdException : Exception
```

VB

```
<SerializableAttribute>  
Public Class InvalidTypeIdException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class InvalidTypeIdException : public Exception
```

F#

```
[<SerializableAttribute>]  
type InvalidTypeIdException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

PropertyTypeInUseException Class

Exception thrown when at least one vertex or edge uses a property type when attempting to remove the property type

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.PropertyTypeInUseException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class PropertyTypeInUseException : Exception
```

VB

```
<SerializableAttribute>  
Public Class PropertyTypeInUseException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class PropertyTypeInUseException : public Exception
```

F#

```
[<SerializableAttribute>]  
type PropertyTypeInUseException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

VertexAlreadyExistException Class

Exception thrown when a vertex id is specified for a vertex type and the vertex already exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.VertexAlreadyExistException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VertexAlreadyExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class VertexAlreadyExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class VertexAlreadyExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type VertexAlreadyExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

VertexDoesNotExistException Class

Exception thrown when a vertex id is specified for a vertex type and the vertex does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.VertexDoesNotExistException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VertexDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class VertexDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class VertexDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type VertexDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

VertexTypeDoesNotExistException Class

Exception thrown when a vertex type id is specified and the vertex type does not exist.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.VertexTypeDoesNotExistException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VertexTypeDoesNotExistException : Exception
```

VB

```
<SerializableAttribute>  
Public Class VertexTypeDoesNotExistException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class VertexTypeDoesNotExistException : public Exception
```

F#

```
[<SerializableAttribute>]  
type VertexTypeDoesNotExistException =  
    class  
        inherit Exception  
    end
```

See Also

[VelocityGraph.Exceptions Namespace](#)

VertexTypeInUseException Class

Exception thrown when at least one vertex exist for a vertex type and attempting to remove the vertex type.

Inheritance Hierarchy

[System.Object](#)

[System.Exception](#)

VelocityGraph.Exceptions.VertexTypeInUseException

Namespace: [VelocityGraph.Exceptions](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]  
public class VertexTypeInUseException : Exception
```

VB

```
<SerializableAttribute>  
Public Class VertexTypeInUseException  
    Inherits Exception
```

C++

```
[SerializableAttribute]  
public ref class VertexTypeInUseException : public Exception
```

F#

```
[<SerializableAttribute>]  
type VertexTypeInUseException =  
    class  
        inherit Exception  
    end
```








See Also

[VelocityGraph.Exceptions Namespace](#)







VelocityGraph.Frontenac.Blueprints Namespace

The **VelocityGraph.Frontenac.Blueprints** namespace contains classes for the VelocityGraph Graph Database.

Classes

| Class | Description |
|---------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CompareHelpers | |
|  DictionaryElement | |
|  Directions | |
|  Features | Features provides a listing of the features/qualities/quirks associated with any Graph implementation. This feature listing can be used to dynamically adjust code to the features of the graph implementation. For example, this feature listing is used extensively throughout the Blueprints TestSuite to validate behavior of the implementation. |
|  GraphHelpers | |
|  Parameter | |
|  Parameter(TK, TV) | |

Interfaces

| Interface | Description |
|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ICloseableIterable(T) | A CloseableIterable is required where it is necessary to deallocate resources from an IEnumerable. |
|  IEdge | An Edge links two vertices. Along with its key/value properties, an edge has both a directionality and a label. The directionality determines which vertex is the tail vertex (out vertex) and which vertex is the head vertex (in vertex). The edge label determines the type of relationship that exists between the two vertices. Diagrammatically, outVertex ---label---> inVertex. |
|  IElement | An Element is the base class for both vertices and edges. An element has an identifier that must be unique to its inheriting classes (vertex or edges). An element can maintain a collection of key/value properties. Keys are always Strings and values can be any object. Particular implementations can reduce the space of objects that can be used as values. Typically, objects are C# primitives (e.g. string, long, int, boolean, etc.) |
|  IGraph | A Graph is a container object for a collection of vertices and a collection edges. |
|  IIndex | |
|  IIndexableGraph | An IIndexableGraph is a graph that supports the manual indexing of its elements. An index is typically some sort of tree structure that allows for the fast lookup of elements by key/value pairs. Indices |

| | | |
|---|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | have an Index object associated with them and allow the user to specify the putting and getting of elements into the index. |
| → | IKeyIndexableGraph | A KeyIndexableGraph is a graph that supports basic index functionality around the key/value pairs of the elements of the graph. By creating key indices for a particular property key, that key is indexed on all the elements of the graph. This has ramifications for quick lookups on methods like <code>getVertices(string, object)</code> and <code>getEdges(string, object)</code> . |
| → | IQuery | |
| → | IThreadedTransactionalGraph | ThreadedTransactionalGraph provides more fine grained control over the transactional context. While TransactionalGraph binds each transaction to the executing thread, ThreadedTransactionalGraph's <code>newTransaction</code> returns a TransactionalGraph that represents its own transactional context independent of the executing thread. Hence, one can have multiple threads operating against a single transaction represented by the returned TransactionalGraph object. This is useful for parallelizing graph algorithms. Note, that one needs to call <code>TransactionalGraph.Commit()</code> or <code>TransactionalGraph.Rollback()</code> to close the transactions returned |
| → | ITransactionalGraph | A transactional graph supports the notion of transactions. A transaction scopes a logically coherent operation composed of multiple read and write operations that either occurs at once or not at all. The exact notion of a transaction and its isolational guarantees depend on the implementing graph database. A transaction scopes a coherent and complete operations. Any element references created during a transaction should not be accessed outside its scope (i.e. after the transaction is committed or rolled back). Accessing such references outside the transactional context they were created in may lead to exceptions. If such access is necessary, the transactional context should be extended. By default, the first operation on a TransactionalGraph will start a transaction automatically. |
| → | IVertex | A vertex maintains pointers to both a set of incoming and outgoing edges. The outgoing edges are those edges for which the vertex is the tail. The incoming edges are those edges for which the vertex is the head. Diagrammatically, <code>---inEdges---</code> vertex <code>---outEdges---</code> . |
| → | IVertexQuery | A VertexQuery object defines a collection of filters and modifiers that are used to intelligently select edges from a vertex. |

Enumerations

| | Enumeration | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | Compare | |
|  | Direction | |

Compare Enumeration

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Compare"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum Compare
```

VB

```
Public Enumeration Compare
```

C++

```
public enum class Compare
```

F#

```
type Compare
```

Members

| Member name | Value | Description |
|------------------|-------|-------------|
| Equal | 0 | |
| NotEqual | 1 | |
| GreaterThan | 2 | |
| GreaterThanEqual | 3 | |
| LessThan | 4 | |
| LessThanEqual | 5 | |
| Within | 6 | |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

CompareHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.CompareHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.CompareHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class CompareHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class CompareHelpers
```

C++

```
[ExtensionAttribute]  
public ref class CompareHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type CompareHelpers = class end
```

The **CompareHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Opposite | |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

CompareHelpers.CompareHelpers Methods

The [CompareHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Opposite | |

See Also

[CompareHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

CompareHelpers.Opposite Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.CompareHelpers.Opposite(VelocityGraph.Frontenac.Blueprints.Compare)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Compare Opposite(  
    this Compare compare  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function Opposite (  
    compare As Compare  
) As Compare
```

C++

```
public:  
[ExtensionAttribute]  
static Compare Opposite(  
    Compare compare  
)
```

F#

```
[<ExtensionAttribute>]  
static member Opposite :  
    compare : Compare -> Compare
```

Parameters

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.CompareHelpers.Opposite(VelocityGraph.Frontenac.Blueprints.Compare)"]

Return Value

Type: [Compare](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.CompareHelpers.Opposite(VelocityGraph.Frontenac.Blueprints.Compare)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Compare](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[CompareHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.DictionaryElement"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.DictionaryElement

[VelocityGraph.Element](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement](#)

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public abstract class DictionaryElement : IElement,
    IDictionary<string, Object>, ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection
```

VB

```
<SerializableAttribute>
Public MustInherit Class DictionaryElement
    Implements IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
[SerializableAttribute]
public ref class DictionaryElement abstract : IElement,
    IDictionary<String^, Object^>, ICollection<KeyValuePair<String^,
Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>, IEnumerable,
    IDictionary, ICollection
```

F#











```
[<AbstractClassAttribute>]
[<SerializableAttribute>]
type DictionaryElement =
    class
        interface IElement
```

```














interface IDictionary<string, Object>
interface ICollection<KeyValuePair<string, Object>>
interface IEnumerable<KeyValuePair<string, Object>>
interface IEnumerable
interface IDictionary
interface ICollection
end
    
```







The **DictionaryElement** type exposes the following members.

Properties




| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Count | |
|  | Graph | |
|  | Id | |
|  | IsFixedSize | |
|  | IsReadOnly | |
|  | IsSynchronized | |
|  | Item | |
|  | Keys | |
|  | SyncRoot | |
|  | Values | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|-------------|
|  | Add(KeyValuePair(String, Object)) | |
|  | Add(Object, Object) | |
|  | Add(String, Object) | |
|  | Clear | |
|  | Contains(KeyValuePair(String, Object)) | |
|  | Contains(Object) | |
|  | ContainsKey | |
|  | CopyTo(Array, Int32) | |
|  | CopyTo(KeyValuePair(String, Object)[], Int32) | |
|  | GetEnumerator | |
|  | GetProperty | |
|  | GetPropertyKeys | |
|  | Remove() | |

| | | |
|-----------------------------------------------------------------------------------|------------------------------------------------------|--|
|  | Remove(KeyValuePair(String, Object)) | |
|  | Remove(Object) | |
|  | Remove(String) | |
|  | RemoveProperty | |
|  | SetProperty | |
|  | TryGetValue | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |











See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.DictionaryElement Properties

The [DictionaryElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Count | |
|  | Graph | |
|  | Id | |
|  | IsFixedSize | |
|  | IsReadOnly | |
|  | IsSynchronized | |
|  | Item | |
|  | Keys | |
|  | SyncRoot | |
|  | Values | |

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Count Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Count"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual int Count { get; }
```

VB

```
Public Overridable ReadOnly Property Count As Integer  
    Get
```

C++

```
public:  
virtual property int Count {  
    int get ();  
}
```

F#

```
abstract Count : int with get  
override Count : int with get
```

Property Value

Type: [Int32](#)

Implements

[ICollection\(T\).Count](#)

[ICollection.Count](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Graph Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Graph"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph Graph { get; }
```

VB

```
Public ReadOnly Property Graph As IGraph  
    Get
```

C++

```
public:  
virtual property IGraph^ Graph {  
    IGraph^ get () sealed;  
}
```

F#

```
abstract Graph : IGraph with get  
override Graph : IGraph with get
```

Property Value

Type: [IGraph](#)

Implements

[IElement.Graph](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Id Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Object Id { get; }
```

VB

```
Public MustOverride ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () abstract;  
}
```

F#

```
abstract Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.IsFixedSize Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.IsFixedSize"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsFixedSize { get; }
```

VB

```
Public ReadOnly Property IsFixedSize As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsFixedSize {  
    bool get () sealed;  
}
```

F#

```
abstract IsFixedSize : bool with get  
override IsFixedSize : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[IDictionary.IsFixedSize](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.IsReadOnly Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.IsReadOnly"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsReadOnly { get; }
```

VB

```
Public ReadOnly Property IsReadOnly As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsReadOnly {  
    bool get () sealed;  
}
```

F#

```
abstract IsReadOnly : bool with get  
override IsReadOnly : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[ICollection\(T\).IsReadOnly](#)

[IDictionary.IsReadOnly](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.IsSynchronized Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.IsSynchronized"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool IsSynchronized { get; }
```

VB

```
Public Overridable ReadOnly Property IsSynchronized As Boolean  
    Get
```

C++

```
public:  
virtual property bool IsSynchronized {  
    bool get ();  
}
```

F#

```
abstract IsSynchronized : bool with get  
override IsSynchronized : bool with get
```

Property Value

Type: [Boolean](#)

Implements

[ICollection.IsSynchronized](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Item Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Item(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Object this[
    string key
] { get; set; }
```

VB

```
Public Overridable Default Property Item (
    key As String
) As Object
    Get
    Set
```

C++

```
public:
virtual property Object^ default[String^ key] {
    Object^ get (String^ key);
    void set (String^ key, Object^ value);
}
```

F#

```
abstract Item : Object with get, set
override Item : Object with get, set
```

Parameters

key

Type: [System.String](#)

Property Value

Type: [Object](#)

Implements

[IDictionary\(TKey, TValue\).Item\(TKey\)](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Keys Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Keys"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ICollection<string> Keys { get; }
```

VB

```
Public Overridable ReadOnly Property Keys As ICollection(Of String)  
    Get
```

C++

```
public:  
virtual property ICollection<String^> Keys {  
    ICollection<String^> get ();  
}
```

F#

```
abstract Keys : ICollection<string> with get  
override Keys : ICollection<string> with get
```

Property Value

Type: [ICollection\(String\)](#)

Implements

[IDictionary\(TKey, TValue\).Keys](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.SyncRoot Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.SyncRoot"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Object SyncRoot { get; }
```

VB

```
Public Overridable ReadOnly Property SyncRoot As Object  
    Get
```

C++

```
public:  
virtual property Object^ SyncRoot {  
    Object^ get ();  
}
```

F#

```
abstract SyncRoot : Object with get  
override SyncRoot : Object with get
```

Property Value

Type: [Object](#)

Implements

[ICollection.SyncRoot](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Values Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Values"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual ICollection<Object> Values { get; }
```

VB

```
Public Overridable ReadOnly Property Values As ICollection(Of Object)  
    Get
```

C++

```
public:  
virtual property ICollection<Object^>^ Values {  
    ICollection<Object^>^ get ();  
}
```

F#

```
abstract Values : ICollection<Object> with get  
override Values : ICollection<Object> with get
```

Property Value

Type: [ICollection\(Object\)](#)

Implements

[IDictionary\(TKey, TValue\).Values](#)

See Also




















[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)




DictionaryElement.DictionaryElement Methods






The [DictionaryElement](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------|-------------|
|  | Add(KeyValuePair(String, Object)) | |
|  | Add(Object, Object) | |
|  | Add(String, Object) | |
|  | Clear | |
|  | Contains(KeyValuePair(String, Object)) | |
|  | Contains(Object) | |
|  | ContainsKey | |
|  | CopyTo(Array, Int32) | |
|  | CopyTo(KeyValuePair(String, Object)[], Int32) | |
|  | GetEnumerator | |
|  | GetProperty | |
|  | GetPropertyKeys | |
|  | Remove() | |
|  | Remove(KeyValuePair(String, Object)) | |
|  | Remove(Object) | |
|  | Remove(String) | |
|  | RemoveProperty | |
|  | SetProperty | |
|  | TryGetValue | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |




See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Add Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------|-------------|
|  | Add(KeyValuePair(String, Object)) | |
|  | Add(Object, Object) | |
|  | Add(String, Object) | |

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Add Method (KeyValuePair(String, Object))

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Add(  
    KeyValuePair<string, Object> item  
)
```

VB

```
Public Overridable Sub Add (  
    item As KeyValuePair(Of String, Object)  
)
```

C++

```
public:  
virtual void Add(  
    KeyValuePair<String^, Object^> item  
)
```

F#

```
abstract Add :  
    item : KeyValuePair<string, Object> -> unit  
override Add :  
    item : KeyValuePair<string, Object> -> unit
```

Parameters

item

Type: [System.Collections.Generic.KeyValuePair\(String, Object\)](#)

[Missing <param name="item"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Implements

[ICollection\(T\).Add\(T\)](#)

See Also

[DictionaryElement Class](#)

[Add Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Add Method (Object, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Add(  
    Object key,  
    Object value  
)
```

VB

```
Public Overridable Sub Add (  
    key As Object,  
    value As Object  
)
```

C++

```
public:  
virtual void Add(  
    Object^ key,  
    Object^ value  
)
```

F#

```
abstract Add :  
    key : Object *  
    value : Object -> unit  
override Add :  
    key : Object *  
    value : Object -> unit
```

Parameters

key

Type: [System.Object](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.Object,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.Object,System.Object)"]

VelocityDB Class Library

Implements

[IDictionary.Add\(Object, Object\)](#)

See Also

[DictionaryElement Class](#)

[Add Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Add Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Add(  
    string key,  
    Object value  
)
```

VB

```
Public Overridable Sub Add (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void Add(  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract Add :  
    key : string *  
    value : Object -> unit  
override Add :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Add(System.String,System.Object)"]

VelocityDB Class Library

Implements

[IDictionary\(TKey, TValue\).Add\(TKey, TValue\)](#)

See Also

[DictionaryElement Class](#)

[Add Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Clear Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Clear"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Clear()
```

VB

```
Public Overridable Sub Clear
```

C++

```
public:  
virtual void Clear()
```

F#

```
abstract Clear : unit -> unit  
override Clear : unit -> unit
```

Implements

[ICollection\(T\).Clear\(\)](#)

[IDictionary.Clear\(\)](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Contains Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------|-------------|
|  | Contains(KeyValuePair(String, Object)) | |
|  | Contains(Object) | |

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Contains Method (KeyValuePair(String, Object))

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    KeyValuePair<string, Object> item  
)
```

VB

```
Public Function Contains (  
    item As KeyValuePair(Of String, Object)  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    KeyValuePair<String^, Object^> item  
) sealed
```

F#

```
abstract Contains :  
    item : KeyValuePair<string, Object> -> bool  
override Contains :  
    item : KeyValuePair<string, Object> -> bool
```

Parameters

item

Type: [System.Collections.Generic.KeyValuePair\(String, Object\)](#)

[Missing <param name="item"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

VelocityDB Class Library

Implements

[ICollection\(T\).Contains\(T\)](#)

See Also

[DictionaryElement Class](#)

[Contains Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Contains Method (Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Object key  
)
```

VB

```
Public Function Contains (  
    key As Object  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Object^ key  
) sealed
```

F#

```
abstract Contains :  
    key : Object -> bool  
override Contains :  
    key : Object -> bool
```

Parameters

key

Type: [System.Object](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Contains(System.Object)"]

Implements

[IDictionary.Contains\(Object\)](#)

VelocityDB Class Library

See Also

[DictionaryElement Class](#)

[Contains Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.ContainsKey Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.ContainsKey(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool ContainsKey(  
    string key  
)
```

VB

```
Public Overridable Function ContainsKey (  
    key As String  
) As Boolean
```

C++

```
public:  
virtual bool ContainsKey(  
    String^ key  
)
```

F#

```
abstract ContainsKey :  
    key : string -> bool  
override ContainsKey :  
    key : string -> bool
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.ContainsKey(System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.ContainsKey(System.String)"]

Implements

[IDictionary\(TKey, TValue\).ContainsKey\(TKey\)](#)

VelocityDB Class Library


See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.CopyTo Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|-------------|
|  | CopyTo(Array, Int32) | |
|  | CopyTo(KeyValuePair(String, Object)[], Int32) | |

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.CopyTo Method (Array, Int32)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Array,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CopyTo(  
    Array array,  
    int index  
)
```

VB

```
Public Sub CopyTo (  
    array As Array,  
    index As Integer  
)
```

C++

```
public:  
virtual void CopyTo(  
    Array^ array,  
    int index  
) sealed
```

F#

```
abstract CopyTo :  
    array : Array *  
    index : int -> unit  
override CopyTo :  
    array : Array *  
    index : int -> unit
```

Parameters

array

Type: [System.Array](#)

[Missing <param name="array"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Array,System.Int32)"]

index

Type: [System.Int32](#)

[Missing <param name="index"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Array,System.Int32)"]

VelocityDB Class Library

Implements

[ICollection.CopyTo\(Array, Int32\)](#)

See Also

[DictionaryElement Class](#)

[CopyTo Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.CopyTo Method (KeyValuePair(String, Object)[], Int32)

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Collections.Generic.KeyValuePair{System.String,System.Object}[],System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void CopyTo(  
    KeyValuePair<string, Object>[] array,  
    int arrayIndex  
)
```

VB

```
Public Overridable Sub CopyTo (  
    array As KeyValuePair(Of String, Object) (),  
    arrayIndex As Integer  
)
```

C++

```
public:  
virtual void CopyTo(  
    array<KeyValuePair<String^, Object^>>^ array,  
    int arrayIndex  
)
```

F#

```
abstract CopyTo :  
    array : KeyValuePair<string, Object>[] *  
    arrayIndex : int -> unit  
override CopyTo :  
    array : KeyValuePair<string, Object>[] *  
    arrayIndex : int -> unit
```

Parameters

array

Type: [System.Collections.Generic.KeyValuePair\(String, Object\)](#)[]

[Missing <param name="array"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Collections.Generic.KeyValuePair{System.String,System.Object}[],System.Int32)"]

arrayIndex

Type: [System.Int32](#)

[Missing <param name="arrayIndex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.CopyTo(System.Collections.Generic.KeyValuePair{System.String,System.Object}[],System.Int32)"]

Implements

[ICollection\(T\).CopyTo\(T\[\], Int32\)](#)

See Also

[DictionaryElement Class](#)

[CopyTo Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<KeyValuePair<string, Object>> GetEnumerator()
```

VB

```
Public Overridable Function GetEnumerator As IEnumerable(Of KeyValuePair(Of String, Object))
```

C++

```
public:  
virtual IEnumerable<KeyValuePair<String^, Object^>> GetEnumerator()
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<KeyValuePair<string, Object>>  
override GetEnumerator : unit -> IEnumerable<KeyValuePair<string, Object>>
```

Return Value

Type: [IEnumerable\(KeyValuePair\(String, Object\)\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Object GetProperty(  
    string key  
)
```

VB

```
Public MustOverride Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) abstract
```

F#

```
abstract GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetProperty(System.String)"]

Implements

[IElement.GetProperty\(String\)](#)

See Also

[DictionaryElement Class](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public MustOverride Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () abstract
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.GetPropertyKeys"]

Implements

[IElement.GetPropertyKeys\(\)](#)





See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Remove Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------------------|-------------|
|  | Remove() | |
|  | Remove(KeyValuePair(String, Object)) | |
|  | Remove(Object) | |
|  | Remove(String) | |

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Remove Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Remove ()
```

VB

```
Public MustOverride Sub Remove
```

C++

```
public:  
virtual void Remove () abstract
```

F#

```
abstract Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[DictionaryElement Class](#)

[Remove Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Remove Method (KeyValuePair(String, Object))

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool Remove(  
    KeyValuePair<string, Object> item  
)
```

VB

```
Public Overridable Function Remove (  
    item As KeyValuePair(Of String, Object)  
) As Boolean
```

C++

```
public:  
virtual bool Remove(  
    KeyValuePair<String^, Object^> item  
)
```

F#

```
abstract Remove :  
    item : KeyValuePair<string, Object> -> bool  
override Remove :  
    item : KeyValuePair<string, Object> -> bool
```

Parameters

item

Type: [System.Collections.Generic.KeyValuePair\(String, Object\)](#)

[Missing <param name="item"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.Collections.Generic.KeyValuePair{System.String,System.Object})"]

VelocityDB Class Library

Implements

[ICollection\(T\).Remove\(T\)](#)

See Also

[DictionaryElement Class](#)

[Remove Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Remove Method (Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove (  
    Object key  
)
```

VB

```
Public Sub Remove (  
    key As Object  
)
```

C++

```
public:  
virtual void Remove (  
    Object^ key  
) sealed
```

F#

```
abstract Remove :  
    key : Object -> unit  
override Remove :  
    key : Object -> unit
```

Parameters

key

Type: [System.Object](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.Object)"]

Implements

[IDictionary.Remove\(Object\)](#)

See Also

[DictionaryElement Class](#)

[Remove Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.Remove Method (String)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool Remove(  
    string key  
)
```

VB

```
Public Overridable Function Remove (  
    key As String  
) As Boolean
```

C++

```
public:  
virtual bool Remove(  
    String^ key  
)
```

F#

```
abstract Remove :  
    key : string -> bool  
override Remove :  
    key : string -> bool
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.String)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.Remove(System.String)"]

Implements

[IDictionary\(TKey, TValue\).Remove\(TKey\)](#)

VelocityDB Class Library

See Also

[DictionaryElement Class](#)

[Remove Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.RemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Object RemoveProperty(  
    string key  
)
```

VB

```
Public MustOverride Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) abstract
```

F#

```
abstract RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.RemoveProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.RemoveProperty(System.String)"]

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[DictionaryElement Class](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.SetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public MustOverride Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) abstract
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.SetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.SetProperty(System.String,System.Object)"]

VelocityDB Class Library

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

DictionaryElement.TryGetValue Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.TryGetValue(System.String,System.Object @)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual bool TryGetValue(  
    string key,  
    out Object value  
)
```

VB

```
Public Overridable Function TryGetValue (  
    key As String,  
    <OutAttribute> ByRef value As Object  
) As Boolean
```

C++

```
public:  
virtual bool TryGetValue(  
    String^ key,  
    [OutAttribute] Object^% value  
)
```

F#

```
abstract TryGetValue :  
    key : string *  
    value : Object byref -> bool  
override TryGetValue :  
    key : string *  
    value : Object byref -> bool
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.TryGetValue(System.String,System.Object @)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.TryGetValue(System.String,System.Object @)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.DictionaryElement.TryGetValue(System.String,System.Object @)"]

Implements

[IDictionary\(TKey, TValue\).TryGetValue\(TKey, TValue\)](#)

See Also

[DictionaryElement Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Direction Enumeration

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Direction"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum Direction
```

VB

```
Public Enumeration Direction
```

C++

```
public enum class Direction
```

F#

```
type Direction
```

Members

| | Member name | Value | Description |
|--|-------------|-------|-------------|
| | Out | 0 | |
| | In | 1 | |
| | Both | 2 | |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Directions Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Directions"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Directions

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class Directions
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class Directions
```

C++

```
[ExtensionAttribute]  
public ref class Directions abstract sealed
```

F#


```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type Directions = class end
```

The **Directions** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Opposite | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-------------|
|  | Proper | |


See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Directions.Directions Methods

The [Directions](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Opposite | |

See Also

[Directions Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Directions.Opposite Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Directions.Opposite(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Direction Opposite(  
    this Direction direction  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function Opposite (  
    direction As Direction  
) As Direction
```

C++

```
public:  
[ExtensionAttribute]  
static Direction Opposite(  
    Direction direction  
)
```

F#

```
[<ExtensionAttribute>]  
static member Opposite :  
    direction : Direction -> Direction
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Directions.Opposite(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [Direction](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Directions.Opposite(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [Direction](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also


[Directions Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Directions.Directions Fields

The [Directions](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Proper | |

See Also

[Directions Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Directions.Proper Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Directions.Proper"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly Direction[] Proper
```

VB

```
Public Shared ReadOnly Proper As Direction()
```

C++

```
public:  
static initonly array<Direction>^ Proper
```

F#

```
static val Proper: Direction[]
```

Field Value

Type: [Direction\[\]](#)

See Also

[Directions Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features Class

Features provides a listing of the features/qualities/quirks associated with any Graph implementation. This feature listing can be used to dynamically adjust code to the features of the graph implementation. For example, this feature listing is used extensively throughout the Blueprints TestSuite to validate behavior of the implementation.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Features

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Features
```

VB

```
Public Class Features
```

C++


```
public ref class Features
```

F#




```
type Features = class end
```
























The **Features** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------|
|  | Features | Initializes a new instance of the Features class |






Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | IgnoresSuppliedIds | Does the graph ignore user provided ids in graph.addVertex(object id)? |
|  | IsPersistent | Does the graph persist the graph to disk after shutdown? |
|  | IsRdfModel | Is the graph an RDF framework? Deprecated thus far, isRdfModel describes a collection of features. Use actual features to describe your data model. |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
|  IsWrapper | Does the graph implement WrapperGraph? |
|  SupportsBooleanProperty | Does the graph allows boolean to be used as a property value for a graph element? |
|  SupportsDoubleProperty | Does the graph allows double to be used as a property value for a graph element? |
|  SupportsDuplicateEdges | Does the graph allow for two edges with the same vertices and edge label to exist? |
|  SupportsEdgeIndex | Does the graph support the indexing of edges by their properties? |
|  SupportsEdgeIteation | Does the graph support graph.getEdges()? |
|  SupportsEdgeKeyIndex | Does the graph support key indexing on edges? |
|  SupportsEdgeProperties | Does the graph support setting and retrieving properties on edges? |
|  SupportsEdgeRetrieval | Does the graph support retrieving edges by id, i.e. graph.getEdge(object id)? |
|  SupportsFloatProperty | Does the graph allows float to be used as a property value for a graph element? |
|  SupportsIdProperty | Does the graph allow the use of the 'id' property name? |
|  SupportsIndices | Does the graph implement IIndexableGraph? |
|  SupportsIntegerProperty | Does the graph allows integer to be used as a property value for a graph element? |
|  SupportsKeyIndices | Does the graph implement KeyIndexableGraph? |
|  SupportsLabelProperty | Does the graph allow the use of the 'label' property name? |
|  SupportsLongProperty | Does the graph allows long to be used as a property value for a graph element? |
|  SupportsMapProperty | Does the graph allows map to be used as a property value for a graph element? |
|  SupportsMixedListProperty | Does the graph allows a mixed list (different data types within the same list) to be used as a property value for a graph element? |
|  SupportsPrimitiveArrayProperty | Does the graph allows a primitive array to be used as a property value for a graph element? |
|  SupportsSelfLoops | Does the graph allow an edge to have the same out/tail and in/head vertex? |
|  SupportsSerializableObjectProperty | Does the graph allow any serializable object to be used as a property value for a graph element? |
|  SupportsStringProperty | Graph allows string to be used as a property value for a graph element. |
|  SupportsThreadedTransactions | Does the graph implement ThreadedTransactionalGraph? |

| | |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|  SupportsTransactions | Does the graph implement TransactionalGraph? |
|  SupportsUniformListProperty | Does the graph allows list (all objects with the list have the same data types) to be used as a property value for a graph element? |
|  SupportsVertexIndex | Does the graph support the indexing of vertices by their properties? |
|  SupportsVertexIteration | Does the graph support graph.getVertices()? |
|  SupportsVertexKeyIndex | Does the graph support key indexing on vertices? |
|  SupportsVertexProperties | Does the graph support setting and retrieving properties on vertices? |

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CheckCompliance | This method determines whether the full gamut of features have been set by the Graph implementation. This is useful for implementers to ensure that they did not miss specifying a feature. Throws <code>InvalidOperationException</code> if a feature was not set |
|  CopyFeatures | This method copies the features in this features object to another feature object. |
|  SupportsElementProperties | Checks whether the graph supports both vertex and edge properties |
|  ToMap | |
|  ToString | (Overrides Object.ToString() .) |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features Constructor

Initializes a new instance of the [Features](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features ()
```

VB

```
Public Sub New
```

C++

```
public:  
Features ()
```

F#

```
new : unit -> Features
```

See Also





















[Features Class](#)













[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.Features Properties

The [Features](#) type exposes the following members.

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|  IgnoresSuppliedIds | Does the graph ignore user provided ids in graph.addVertex(object id)? |
|  IsPersistent | Does the graph persist the graph to disk after shutdown? |
|  IsRdfModel | Is the graph an RDF framework? Deprecated thus far, isRdfModel describes a collection of features. Use actual features to describe your data model. |
|  IsWrapper | Does the graph implement WrapperGraph? |
|  SupportsBooleanProperty | Does the graph allows boolean to be used as a property value for a graph element? |
|  SupportsDoubleProperty | Does the graph allows double to be used as a property value for a graph element? |
|  SupportsDuplicateEdges | Does the graph allow for two edges with the same vertices and edge label to exist? |
|  SupportsEdgeIndex | Does the graph support the indexing of edges by their properties? |
|  SupportsEdgeIiteration | Does the graph support graph.getEdges()? |
|  SupportsEdgeKeyIndex | Does the graph support key indexing on edges? |
|  SupportsEdgeProperties | Does the graph support setting and retrieving properties on edges? |
|  SupportsEdgeRetrieval | Does the graph support retrieving edges by id, i.e. graph.getEdge(object id)? |
|  SupportsFloatProperty | Does the graph allows float to be used as a property value for a graph element? |
|  SupportsIdProperty | Does the graph allow the use of the 'id' property name? |
|  SupportsIndices | Does the graph implement IIndexableGraph? |
|  SupportsIntegerProperty | Does the graph allows integer to be used as a property value for a graph element? |
|  SupportsKeyIndices | Does the graph implement KeyIndexableGraph? |
|  SupportsLabelProperty | Does the graph allow the use of the 'label' property name? |
|  SupportsLongProperty | Does the graph allows long to be used as a property value for a graph element? |
|  SupportsMapProperty | Does the graph allows map to be used as a property value for a graph element? |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|  SupportsMixedListProperty | Does the graph allows a mixed list (different data types within the same list) to be used as a property value for a graph element? |
|  SupportsPrimitiveArrayProperty | Does the graph allows a primitive array to be used as a property value for a graph element? |
|  SupportsSelfLoops | Does the graph allow an edge to have the same out/tail and in/head vertex? |
|  SupportsSerializableObjectProperty | Does the graph allow any serializable object to be used as a property value for a graph element? |
|  SupportsStringProperty | Graph allows string to be used as a property value for a graph element. |
|  SupportsThreadedTransactions | Does the graph implement ThreadedTransactionalGraph? |
|  SupportsTransactions | Does the graph implement TransactionalGraph? |
|  SupportsUniformListProperty | Does the graph allows list (all objects with the list have the same data types) to be used as a property value for a graph element? |
|  SupportsVertexIndex | Does the graph support the indexing of vertices by their properties? |
|  SupportsVertexIteration | Does the graph support graph.getVertices()? |
|  SupportsVertexKeyIndex | Does the graph support key indexing on vertices? |
|  SupportsVertexProperties | Does the graph support setting and retrieving properties on vertices? |

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.IgnoresSuppliedIds Property

Does the graph ignore user provided ids in graph.addVertex(object id)?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IgnoresSuppliedIds { get; set; }
```

VB

```
Public Property IgnoresSuppliedIds As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool IgnoresSuppliedIds {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member IgnoresSuppliedIds : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.IsPersistent Property

Does the graph persist the graph to disk after shutdown?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsPersistent { get; set; }
```

VB

```
Public Property IsPersistent As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool IsPersistent {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member IsPersistent : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.IsRdfModel Property

Is the graph an RDF framework? Deprecated thus far, isRDFModel describes a collection of features. Use actual features to describe your data model.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsRdfModel { get; set; }
```

VB

```
Public Property IsRdfModel As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool IsRdfModel {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member IsRdfModel : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.IsWrapper Property

Does the graph implement WrapperGraph?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsWrapper { get; set; }
```

VB

```
Public Property IsWrapper As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool IsWrapper {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member IsWrapper : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsBooleanProperty Property

Does the graph allows boolean to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsBooleanProperty { get; set; }
```

VB

```
Public Property SupportsBooleanProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsBooleanProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsBooleanProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsDoubleProperty Property

Does the graph allows double to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsDoubleProperty { get; set; }
```

VB

```
Public Property SupportsDoubleProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsDoubleProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsDoubleProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsDuplicateEdges Property

Does the graph allow for two edges with the same vertices and edge label to exist?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsDuplicateEdges { get; set; }
```

VB

```
Public Property SupportsDuplicateEdges As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsDuplicateEdges {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsDuplicateEdges : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsEdgeIndex Property

Does the graph support the indexing of edges by their properties?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsEdgeIndex { get; set; }
```

VB

```
Public Property SupportsEdgeIndex As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsEdgeIndex {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsEdgeIndex : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsEdgeIteration Property

Does the graph support graph.getEdges()?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsEdgeIteration { get; set; }
```

VB

```
Public Property SupportsEdgeIteration As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsEdgeIteration {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsEdgeIteration : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsEdgeKeyIndex Property

Does the graph support key indexing on edges?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsEdgeKeyIndex { get; set; }
```

VB

```
Public Property SupportsEdgeKeyIndex As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsEdgeKeyIndex {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsEdgeKeyIndex : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsEdgeProperties Property

Does the graph support setting and retrieving properties on edges?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsEdgeProperties { get; set; }
```

VB

```
Public Property SupportsEdgeProperties As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsEdgeProperties {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsEdgeProperties : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsEdgeRetrieval Property

Does the graph support retrieving edges by id, i.e. graph.getEdge(object id)?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsEdgeRetrieval { get; set; }
```

VB

```
Public Property SupportsEdgeRetrieval As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsEdgeRetrieval {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsEdgeRetrieval : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsFloatProperty Property

Does the graph allows float to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsFloatProperty { get; set; }
```

VB

```
Public Property SupportsFloatProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsFloatProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsFloatProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsIdProperty Property

Does the graph allow the use of the 'id' property name?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsIdProperty { get; set; }
```

VB

```
Public Property SupportsIdProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsIdProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsIdProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsIndices Property

Does the graph implement IIndexableGraph?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsIndices { get; set; }
```

VB

```
Public Property SupportsIndices As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsIndices {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsIndices : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsIntegerProperty Property

Does the graph allows integer to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsIntegerProperty { get; set; }
```

VB

```
Public Property SupportsIntegerProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsIntegerProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsIntegerProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsKeyIndices Property

Does the graph implement KeyIndexableGraph?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsKeyIndices { get; set; }
```

VB

```
Public Property SupportsKeyIndices As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsKeyIndices {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsKeyIndices : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsLabelProperty Property

Does the graph allow the use of the 'label' property name?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsLabelProperty { get; set; }
```

VB

```
Public Property SupportsLabelProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsLabelProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsLabelProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsLongProperty Property

Does the graph allows long to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsLongProperty { get; set; }
```

VB

```
Public Property SupportsLongProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsLongProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsLongProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsMapProperty Property

Does the graph allows map to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsMapProperty { get; set; }
```

VB

```
Public Property SupportsMapProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsMapProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsMapProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsMixedListProperty Property

Does the graph allows a mixed list (different data types within the same list) to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsMixedListProperty { get; set; }
```

VB

```
Public Property SupportsMixedListProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsMixedListProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsMixedListProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsPrimitiveArrayProperty Property

Does the graph allows a primitive array to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsPrimitiveArrayProperty { get; set; }
```

VB

```
Public Property SupportsPrimitiveArrayProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsPrimitiveArrayProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsPrimitiveArrayProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsSelfLoops Property

Does the graph allow an edge to have the same out/tail and in/head vertex?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsSelfLoops { get; set; }
```

VB

```
Public Property SupportsSelfLoops As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsSelfLoops {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsSelfLoops : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsSerializableObjectProperty Property

Does the graph allow any serializable object to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsSerializableObjectProperty { get; set; }
```

VB

```
Public Property SupportsSerializableObjectProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsSerializableObjectProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsSerializableObjectProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsStringProperty Property

Graph allows string to be used as a property value for a graph element.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsStringProperty { get; set; }
```

VB

```
Public Property SupportsStringProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsStringProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsStringProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsThreadedTransactions Property

Does the graph implement ThreadedTransactionalGraph?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsThreadedTransactions { get; set; }
```

VB

```
Public Property SupportsThreadedTransactions As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsThreadedTransactions {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsThreadedTransactions : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsTransactions Property

Does the graph implement TransactionalGraph?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsTransactions { get; set; }
```

VB

```
Public Property SupportsTransactions As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsTransactions {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsTransactions : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsUniformListProperty Property

Does the graph allows list (all objects with the list have the same data types) to be used as a property value for a graph element?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsUniformListProperty { get; set; }
```

VB

```
Public Property SupportsUniformListProperty As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsUniformListProperty {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsUniformListProperty : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsVertexIndex Property

Does the graph support the indexing of vertices by their properties?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsVertexIndex { get; set; }
```

VB

```
Public Property SupportsVertexIndex As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsVertexIndex {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsVertexIndex : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsVertexIteration Property

Does the graph support graph.getVertices()?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsVertexIteration { get; set; }
```

VB

```
Public Property SupportsVertexIteration As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsVertexIteration {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsVertexIteration : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsVertexKeyIndex Property

Does the graph support key indexing on vertices?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsVertexKeyIndex { get; set; }
```

VB

```
Public Property SupportsVertexKeyIndex As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsVertexKeyIndex {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsVertexKeyIndex : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsVertexProperties Property

Does the graph support setting and retrieving properties on vertices?

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsVertexProperties { get; set; }
```

VB

```
Public Property SupportsVertexProperties As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SupportsVertexProperties {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SupportsVertexProperties : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also






[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.Features Methods

The [Features](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | CheckCompliance | This method determines whether the full gamut of features have been set by the Graph implementation. This is useful for implementers to ensure that they did not miss specifying a feature. Throws <code>InvalidOperationException</code> if a feature was not set |
|  | CopyFeatures | This method copies the features in this features object to another feature object. |
|  | SupportsElementProperties | Checks whether the graph supports both vertex and edge properties |
|  | ToMap | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.CheckCompliance Method

This method determines whether the full gamut of features have been set by the Graph implementation. This is useful for implementers to ensure that they did not miss specifying a feature. Throws `InvalidOperationException` if a feature was not set

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CheckCompliance ()
```

VB

```
Public Sub CheckCompliance
```

C++

```
public:  
void CheckCompliance ()
```

F#

```
member CheckCompliance : unit -> unit
```

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.CopyFeatures Method

This method copies the features in this features object to another feature object.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features CopyFeatures ()
```

VB

```
Public Function CopyFeatures As Features
```

C++

```
public:  
Features^ CopyFeatures ()
```

F#

```
member CopyFeatures : unit -> Features
```

Return Value

Type: [Features](#)

a feature object with a clone of the features in the prior.

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.SupportsElementProperties Method

Checks whether the graph supports both vertex and edge properties

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SupportsElementProperties ()
```

VB

```
Public Function SupportsElementProperties As Boolean
```

C++

```
public:  
bool SupportsElementProperties ()
```

F#

```
member SupportsElementProperties : unit -> bool
```

Return Value

Type: [Boolean](#)

whether the graph supports both vertex and edge properties

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.ToMap Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Features.ToMap"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IDictionary<string, Object> ToMap()
```

VB

```
Public Function ToMap As IDictionary(Of String, Object)
```

C++

```
public:  
IDictionary<String^, Object^>^ ToMap()
```

F#

```
member ToMap : unit -> IDictionary<string, Object>
```

Return Value

Type: [IDictionary\(String, Object\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Features.ToMap"]

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Features.ToString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Features.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Features.ToString"]

See Also

[Features Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.GraphHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.GraphHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class GraphHelpers
```

C++










```
[ExtensionAttribute]  
public ref class GraphHelpers abstract sealed
```

F#




```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type GraphHelpers = class end
```

The **GraphHelpers** type exposes the following members.


Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------|-------------|
|  | CreateTinkerGraph | |
|  | Get(TValue)(IList(TValue), Int32) | |
|  | Get(TKey, TValue)(IDictionary(TKey, TValue), TKey) | |
|  | IsNumber | |
|  | JavaRemove(TKey, TValue) | |
|  | LoadDotNet | |
|  | LoadGml | |
|  | LoadGraphml | |
|  | LoadGraphson | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|-----------------------------------|--|
|  | Put(TKey, TValue) | |
|  | SaveDotNet | |
|  | SaveGml | |
|  | SaveGraphml | |
|  | SaveGraphson | |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|-------------|
|  | ContractExceptionName | |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.GraphHelpers Methods

The [GraphHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|--|--------------------------------------------------------------------|-------------|
| | CreateTinkerGraph | |
| | Get(TValue)(IList(TValue), Int32) | |
| | Get(TKey, TValue)(IDictionary(TKey, TValue), TKey) | |
| | IsNumber | |
| | JavaRemove(TKey, TValue) | |
| | LoadDotNet | |
| | LoadGml | |
| | LoadGraphml | |
| | LoadGraphson | |
| | Put(TKey, TValue) | |
| | SaveDotNet | |
| | SaveGml | |
| | SaveGraphml | |
| | SaveGraphson | |

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.CreateTinkerGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.CreateTinkerGraph(VelocityGraph.Frontenac.Blueprints.IGraph)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void CreateTinkerGraph(  
    this IGraph graph  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub CreateTinkerGraph (  
    graph As IGraph  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void CreateTinkerGraph(  
    IGraph^ graph  
)
```

F#

```
[<ExtensionAttribute>]  
static member CreateTinkerGraph :  
    graph : IGraph -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.CreateTinkerGraph(VelocityGraph.Frontenac.Blueprints.IGraph)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

VelocityDB Class Library

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.Get Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------|-------------|
|  | Get(TKey, TValue)(IDictionary(TKey, TValue), TKey) | |
|  | Get(TValue)(IList(TValue), Int32) | |

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.Get(TKey, TValue) Method (IDictionary(TKey, TValue), TKey)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TValue Get<TKey, TValue>(
    this IDictionary<TKey, TValue> dictionary,
    TKey key
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Get(Of TKey, TValue) (
    dictionary As IDictionary(Of TKey, TValue),
    key As TKey
) As TValue
```

C++

```
public:
[ExtensionAttribute]
generic<typename TKey, typename TValue>
static TValue Get(
    IDictionary<TKey, TValue>^ dictionary,
    TKey key
)
```

F#

```
[<ExtensionAttribute>]
static member Get :
    dictionary : IDictionary<'TKey, 'TValue> *
    key : 'TKey -> 'TValue
```

Parameters

dictionary

Type: [System.Collections.Generic.IDictionary\(TKey, TValue\)](#)

[Missing <param name="dictionary"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

key

Type: **TKey**

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Type Parameters

TKey

[Missing <typeparam name="TKey"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

TValue

[Missing <typeparam name="TValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Return Value

Type: **TValue**

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IDictionary\(TKey, TValue\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[Get Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.Get(TValue) Method (IList(TValue), Int32)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`1(System.Collections.Generic.IList{`0},System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TValue Get<TValue>(
    this IList<TValue> list,
    int at
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Get(Of TValue) (
    list As IList(Of TValue),
    at As Integer
) As TValue
```

C++

```
public:
[ExtensionAttribute]
generic<typename TValue>
static TValue Get(
    IList<TValue>^ list,
    int at
)
```

F#

```
[<ExtensionAttribute>]
static member Get :
    list : IList<'TValue> *
    at : int -> 'TValue
```

Parameters

list

Type: [System.Collections.Generic.IList\(TValue\)](#)

[Missing <param name="list"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`1(System.Collections.Generic.IList{`0},System.Int32)"]

at

Type: [System.Int32](#)

[Missing <param name="at"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`1(System.Collections.Generic.IList{`0},System.Int32)"]

Type Parameters

TValue

[Missing <typeparam name="TValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`1(System.Collections.Generic.IList{`0},System.Int32)"]

Return Value

Type: **TValue**

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Get`1(System.Collections.Generic.IList{`0},System.Int32)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IList\(TValue\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[Get Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.IsNumber Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.IsNumber(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool IsNumber(  
    Object expression  
)
```

VB

```
Public Shared Function IsNumber (  
    expression As Object  
) As Boolean
```

C++

```
public:  
static bool IsNumber(  
    Object^ expression  
)
```

F#

```
static member IsNumber :  
    expression : Object -> bool
```

Parameters

expression

Type: [System.Object](#)

[Missing <param name="expression"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.IsNumber(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.IsNumber(System.Object)"]

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.JavaRemove(TKey, TValue) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove``2(System.Collections.Generic.IDictionary{``0,``1},``0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TValue JavaRemove<TKey, TValue>(
    this IDictionary<TKey, TValue> dictionary,
    TKey key
)
```

VB

```
<ExtensionAttribute>
Public Shared Function JavaRemove(Of TKey, TValue) (
    dictionary As IDictionary(Of TKey, TValue),
    key As TKey
) As TValue
```

C++

```
public:
[ExtensionAttribute]
generic<typename TKey, typename TValue>
static TValue JavaRemove(
    IDictionary<TKey, TValue>^ dictionary,
    TKey key
)
```

F#

```
[<ExtensionAttribute>]
static member JavaRemove :
    dictionary : IDictionary<'TKey, 'TValue> *
    key : 'TKey -> 'TValue
```

Parameters

dictionary

Type: [System.Collections.Generic.IDictionary](#)(*TKey*, *TValue*)

[Missing <param name="dictionary"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove``2(System.Collections.Generic.IDictionary{``0,``1},``0)"]

key

Type: **TKey**

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Type Parameters

TKey

[Missing <typeparam name="TKey"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

TValue

[Missing <typeparam name="TValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Return Value

Type: **TValue**

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.JavaRemove`2(System.Collections.Generic.IDictionary{`0,`1},`0)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IDictionary\(TKey, TValue\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.LoadDotNet Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadDotNet(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IGraph LoadDotNet(  
    string directory  
)
```

VB

```
Public Shared Function LoadDotNet (  
    directory As String  
) As IGraph
```

C++

```
public:  
static IGraph^ LoadDotNet(  
    String^ directory  
)
```

F#

```
static member LoadDotNet :  
    directory : string -> IGraph
```

Parameters

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadDotNet(System.String)"]

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadDotNet(System.String)"]

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.LoadGml Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void LoadGml (  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub LoadGml (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void LoadGml (  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member LoadGml :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.LoadGraphml Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void LoadGraphml (  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub LoadGraphml (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void LoadGraphml (  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member LoadGraphml :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.LoadGraphson Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void LoadGraphson(  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub LoadGraphson (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void LoadGraphson(  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member LoadGraphson :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.LoadGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.Put(TKey, TValue) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put`2(System.Collections.Generic.IDictionary{`0,`1},`0,`1)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TValue Put<TKey, TValue>(
    this IDictionary<TKey, TValue> dictionary,
    TKey key,
    TValue value
)
```

VB

```
<ExtensionAttribute>
Public Shared Function Put(Of TKey, TValue) (
    dictionary As IDictionary(Of TKey, TValue),
    key As TKey,
    value As TValue
) As TValue
```

C++

```
public:
[ExtensionAttribute]
generic<typename TKey, typename TValue>
static TValue Put(
    IDictionary<TKey, TValue>^ dictionary,
    TKey key,
    TValue value
)
```

F#

```
[<ExtensionAttribute>]
static member Put :
    dictionary : IDictionary<'TKey, 'TValue> *
    key : 'TKey *
    value : 'TValue -> 'TValue
```

Parameters

dictionary

Type: [System.Collections.Generic.IDictionary\(TKey, TValue\)](#)

[Missing <param name="dictionary"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

key

Type: **TKey**

[Missing <param name="key"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

value

Type: **TValue**

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

Type Parameters

TKey

[Missing <typeparam name="TKey"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

TValue

[Missing <typeparam name="TValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

Return Value

Type: **TValue**

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.Put<<2(System.Collections.Generic.IDictionary{
`0,`1},`0,`1)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IDictionary\(TKey, TValue\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.SaveDotNet Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveDotNet(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SaveDotNet (  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SaveDotNet (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SaveDotNet (  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member SaveDotNet :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveDotNet(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveDotNet(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.SaveGml Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SaveGml (  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SaveGml (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SaveGml (  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member SaveGml :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.SaveGraphml Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SaveGraphml (  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SaveGraphml (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SaveGraphml (  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member SaveGraphml :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphml(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.SaveGraphson Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SaveGraphson(  
    this IGraph graph,  
    string directory  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SaveGraphson (  
    graph As IGraph,  
    directory As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SaveGraphson(  
    IGraph^ graph,  
    String^ directory  
)
```

F#

```
[<ExtensionAttribute>]  
static member SaveGraphson :  
    graph : IGraph *  
    directory : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.GraphHelpers.SaveGraphson(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.GraphHelpers Fields

The [GraphHelpers](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|-------------|
|  | ContractExceptionName | |

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

GraphHelpers.ContractExceptionName Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.GraphHelpers.ContractExceptionName"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string ContractExceptionName =  
"System.Diagnostics.Contracts.__ContractsRuntime+ContractException"
```

VB

```
Public Const ContractExceptionName As String =  
"System.Diagnostics.Contracts.__ContractsRuntime+ContractException"
```

C++

```
public:  
literal String^ ContractExceptionName =  
"System.Diagnostics.Contracts.__ContractsRuntime+ContractException"
```

F#

```
static val mutable ContractExceptionName: string
```

Field Value

Type: [String](#)

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

ICloseableIterable(T) Interface

A CloseableIterable is required where it is necessary to deallocate resources from an IEnumerable.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface ICloseableIterable<out T> : IDisposable,  
    IEnumerable<T>, IEnumerable
```

VB

```
Public Interface ICloseableIterable (Of Out T)  
    Inherits IDisposable, IEnumerable (Of T), IEnumerable
```

C++

```
generic<typename T>  
public interface class ICloseableIterable : IDisposable,  
    IEnumerable<T>, IEnumerable
```

F#

```
type ICloseableIterable<'T> =  
    interface  
        interface IDisposable  
        interface IEnumerable<'T>  
        interface IEnumerable  
    end
```

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "T:VelocityGraph.Frontenac.Blueprints.ICloseableIterable`1"]

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IEdge Interface

An Edge links two vertices. Along with its key/value properties, an edge has both a directionality and a label. The directionality determines which vertex is the tail vertex (out vertex) and which vertex is the head vertex (in vertex). The edge label determines the type of relationship that exists between the two vertices. Diagrammatically, outVertex ---label---> inVertex.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IEdge : IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Interface IEdge
    Inherits IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++


```
public interface class IEdge : IElement,
    IDictionary<String^, Object^>, ICollection<KeyValuePair<String^,
Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>, IEnumerable,
    IDictionary, ICollection
```

F#


```
type IEdge =
    interface
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
        interface ICollection
    end
```

The **IEdge** type exposes the following members.











Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|--------------------------------------------|
|  | Label | Return the label associated with the edge. |

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  GetVertex | Return the tail/out or head/in vertex. ArgumentException is thrown if a direction of both is provided |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  EdgeString | (Defined by StringFactory.) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |

VelocityDB Class Library


See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IEdge.IEdge Properties

The [IEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|--------------------------------------------|
|  | Label | Return the label associated with the edge. |

See Also

[IEdge Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IEdge.Label Property

Return the label associated with the edge.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
string Label { get; }
```

VB

```
ReadOnly Property Label As String  
    Get
```

C++

```
property String^ Label {  
    String^ get ();  
}
```

F#

```
abstract Label : string with get
```

Return Value

Type: [String](#)

the edge label

See Also


[IEdge Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)











IEdge.IEdge Methods

The [IEdge](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  GetVertex | Return the tail/out or head/in vertex. ArgumentException is thrown if a direction of both is provided |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  EdgeString | (Defined by StringFactory.) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |

VelocityDB Class Library

See Also

[IEdge Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IEdge.GetVertex Method

Return the tail/out or head/in vertex. ArgumentException is thrown if a direction of both is provided

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
IVertex^ GetVertex(  
    Direction direction  
)
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

whether to return the tail/out or head/in vertex

Return Value

Type: [IVertex](#)

the tail/out or head/in vertex

See Also

[IEdge Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement Interface

An Element is the base class for both vertices and edges. An element has an identifier that must be unique to its inheriting classes (vertex or edges). An element can maintain a collection of key/value properties. Keys are always Strings and values can be any object. Particular implementations can reduce the space of objects that can be used as values. Typically, objects are C# primitives (e.g. string, long, int, boolean, etc.)

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IElement : IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Interface IElement
    Inherits IDictionary(Of String, Object), ICollection(Of KeyValuePair(Of
String, Object)),
    IEnumerable(Of KeyValuePair(Of String, Object)), IEnumerable,
    IDictionary, ICollection
```

C++


```
public interface class IElement : IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```


F#

```
type IElement =
    interface
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
        interface ICollection
    end
```






The **IElement** type exposes the following members.

Properties









| Name | Description |
|-----------------------------------------------------------------------------------------------------------|------------------------------------|
|  Graph | The graph that owns this IElement. |

| | |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. |
|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  GetProperty | Return the object value associated with the provided string key. If no value exists for that key, return null. |
|  GetPropertyKeys | Return all the keys associated with the element. |
|  Remove | Remove the element from the graph. |
|  RemoveProperty | Un-assigns a key/value property from the element. The object value of the removed property is returned. |
|  SetProperty | Assign a key/value property to the element. If a value already exists for this key, then the previous key/value is overwritten. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are <code>equal()</code> . (Defined by ElementHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be <code>equals()</code> . (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

VelocityDB Class Library

| | |
|--|--------------------------------------------------------------------------------------------------------------------------------------------|
| | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|--|--------------------------------------------------------------------------------------------------------------------------------------------|



See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.IElement Properties

The [IElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Graph | The graph that owns this IElement. |
|  | Id | An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers. |

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.Graph Property

The graph that owns this IElement.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IGraph Graph { get; }
```

VB

```
ReadOnly Property Graph As IGraph  
    Get
```

C++

```
property IGraph^ Graph {  
    IGraph^ get ();  
}
```

F#

```
abstract Graph : IGraph with get
```

Property Value

Type: [IGraph](#)

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.Id Property

An identifier that is unique to its inheriting class. All vertices of a graph must have unique identifiers. All edges of a graph must have unique identifiers.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object Id { get; }
```

VB

```
ReadOnly Property Id As Object  
    Get
```

C++

```
property Object^ Id {  
    Object^ get ();  
}
```

F#

```
abstract Id : Object with get
```

Property Value

Type: [Object](#)

the identifier of the element

See Also






[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)









IElement.IElement Methods

The [IElement](#) type exposes the following members.

Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
|  GetProperty | Return the object value associated with the provided string key. If no value exists for that key, return null. |
|  GetPropertyKeys | Return all the keys associated with the element. |
|  Remove | Remove the element from the graph. |
|  RemoveProperty | Un-assigns a key/value property from the element. The object value of the removed property is returned. |
|  SetProperty | Assign a key/value property to the element. If a value already exists for this key, then the previous key/value is overwritten. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

| | | |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------|

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.GetProperty Method

Return the object value associated with the provided string key. If no value exists for that key, return null.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object GetProperty(  
    string key  
)
```

VB

```
Function GetProperty (  
    key As String  
) As Object
```

C++

```
Object^ GetProperty(  
    String^ key  
)
```

F#

```
abstract GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the key/value property

Return Value

Type: [Object](#)

the object value related to the string key

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.GetPropertyKeys Method

Return all the keys associated with the element.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<string> GetPropertyKeys ()
```

VB

```
Function GetPropertyKeys As IEnumerable (Of String)
```

C++

```
IEnumerable<String^>^ GetPropertyKeys ()
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

the set of all string keys associated with the element

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.Remove Method

Remove the element from the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Remove ()
```

VB

```
Sub Remove
```

C++

```
void Remove ()
```

F#

```
abstract Remove : unit -> unit
```

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.RemoveProperty Method

Un-assigns a key/value property from the element. The object value of the removed property is returned.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object RemoveProperty(  
    string key  
)
```

VB

```
Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
Object^ RemoveProperty(  
    String^ key  
)
```

F#

```
abstract RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

the key of the property to remove from the element

Return Value

Type: [Object](#)

the object value associated with that key prior to removal

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IElement.SetProperty Method

Assign a key/value property to the element. If a value already exists for this key, then the previous key/value is overwritten.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
void SetProperty(  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

the string key of the property

value

Type: [System.Object](#)

the object value o the property

See Also

[IElement Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph Interface

A Graph is a container object for a collection of vertices and a collection edges.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IGraph
```

VB

```
Public Interface IGraph
```

C++


```
public interface class IGraph
```

F#






```
type IGraph = interface end
```








The **IGraph** type exposes the following members.

Properties










| Name | Description |
|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Features | Get the particular features of the graph implementation. Not all graph implementations are identical nor perfectly implement the Blueprints API. The Features object returned contains meta-data about numerous potential divergences between implementations. |

Methods




| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation. |
|  AddVertex | Create a new vertex, add it to the graph, and return the newly created vertex. The provided object identifier is a recommendation for the identifier to use. It is not required that the implementation use this identifier. |
|  GetEdge | Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null. |
|  GetEdges() | Return an iterable to all the edges in the graph. If this is not possible for the implementation, then an NotSupportedException can be thrown. |
|  GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. If this is not possible for the implementation, then an |

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | NotSupportedException can be thrown. The graph implementation should use indexing structures to make this efficient else a full edge-filter scan is required. |
|  | GetVertex | Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null. |
|  | GetVertices() | Return an iterable to all the vertices in the graph. If this is not possible for the implementation, then an NotSupportedException can be thrown. |
|  | GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. If this is not possible for the implementation, then a NotSupportedException can be thrown. The graph implementation should use indexing structures to make this efficient else a full vertex-filter scan is required. |
|  | Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph. |
|  | RemoveEdge | Remove the provided edge from the graph. |
|  | RemoveVertex | Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well. |
|  | Shutdown | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------|
| | | mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |


See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.IGraph Properties

The [IGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Features | Get the particular features of the graph implementation. Not all graph implementations are identical nor perfectly implement the Blueprints API. The Features object returned contains meta-data about numerous potential divergences between implementations. |

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.Features Property

Get the particular features of the graph implementation. Not all graph implementations are identical nor perfectly implement the Blueprints API. The Features object returned contains meta-data about numerous potential divergences between implementations.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Features Features { get; }
```

VB

```
ReadOnly Property Features As Features  
    Get
```

C++

```
property Features^ Features {  
    Features^ get ();  
}
```

F#

```
abstract Features : Features with get
```

Property Value

Type: [Features](#)

the features of this particular Graph implementation

See Also











[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)


IGraph.IGraph Methods













The [IGraph](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation. |
|  AddVertex | Create a new vertex, add it to the graph, and return the newly created vertex. The provided object identifier is a recommendation for the identifier to use. It is not required that the implementation use this identifier. |
|  GetEdge | Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null. |
|  GetEdges() | Return an iterable to all the edges in the graph. If this is not possible for the implementation, then an <code>NotSupportedException</code> can be thrown. |
|  GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. If this is not possible for the implementation, then an <code>NotSupportedException</code> can be thrown. The graph implementation should use indexing structures to make this efficient else a full edge-filter scan is required. |
|  GetVertex | Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null. |
|  GetVertices() | Return an iterable to all the vertices in the graph. If this is not possible for the implementation, then an <code>NotSupportedException</code> can be thrown. |
|  GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. If this is not possible for the implementation, then a <code>NotSupportedException</code> can be thrown. The graph implementation should use indexing structures to make this efficient else a full vertex-filter scan is required. |
|  Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph. |
|  RemoveEdge | Remove the provided edge from the graph. |
|  RemoveVertex | Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well. |
|  Shutdown | |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.AddEdge Method

Add an edge to the graph. The added edges requires a recommended identifier, a tail vertex, an head vertex, and a label. Like adding a vertex, the provided object identifier may be ignored by the implementation.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEdge AddEdge (  
    Object id,  
    IVertex outVertex,  
    IVertex inVertex,  
    string label  
)
```

VB

```
Function AddEdge (  
    id As Object,  
    outVertex As IVertex,  
    inVertex As IVertex,  
    label As String  
) As IEdge
```

C++

```
IEdge^ AddEdge (  
    Object^ id,  
    IVertex^ outVertex,  
    IVertex^ inVertex,  
    String^ label  
)
```

F#

```
abstract AddEdge :  
    id : Object *  
    outVertex : IVertex *  
    inVertex : IVertex *  
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

the recommended object identifier

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

VelocityDB Class Library

the vertex on the tail of the edge

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex on the head of the edge

label

Type: [System.String](#)

the label associated with the edge

Return Value

Type: [IEdge](#)

the newly created edge

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.AddVertex Method

Create a new vertex, add it to the graph, and return the newly created vertex. The provided object identifier is a recommendation for the identifier to use. It is not required that the implementation use this identifier.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertex AddVertex(  
    Object id  
)
```

VB

```
Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
IVertex^ AddVertex(  
    Object^ id  
)
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

the recommended object identifier

Return Value

Type: [IVertex](#)

the newly created vertex

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetEdge Method

Return the edge referenced by the provided object identifier. If no edge is referenced by that identifier, then return null.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEdge GetEdge(  
    Object id  
)
```

VB

```
Function GetEdge (  
    id As Object  
) As IEdge
```

C++

```
IEdge^ GetEdge(  
    Object^ id  
)
```

F#

```
abstract GetEdge :  
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

the identifier of the edge to retrieved from the graph

Return Value

Type: [IEdge](#)

the edge referenced by the provided identifier or null when no such edge exists



See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetEdges() | Return an iterable to all the edges in the graph. If this is not possible for the implementation, then an NotSupportedException can be thrown. |
|  | GetEdges(String, Object) | Return an iterable to all the edges in the graph that have a particular key/value property. If this is not possible for the implementation, then an NotSupportedException can be thrown. The graph implementation should use indexing structures to make this efficient else a full edge-filter scan is required. |

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetEdges Method

Return an iterable to all the edges in the graph. If this is not possible for the implementation, then an `NotSupportedException` can be thrown.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IEdge> GetEdges ()
```

VB

```
Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
IEnumerable<IEdge^>^ GetEdges ()
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

an iterable reference to all edges in the graph

See Also

[IGraph Interface](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetEdges Method (String, Object)

Return an iterable to all the edges in the graph that have a particular key/value property. If this is not possible for the implementation, then an `NotSupportedException` can be thrown. The graph implementation should use indexing structures to make this efficient else a full edge-filter scan is required.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

the key of the edge

value

Type: [System.Object](#)

the value of the edge

Return Value

Type: [IEnumerable\(IEdge\)](#)

an iterable of edges with provided key and value

VelocityDB Class Library

See Also

[IGraph Interface](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetVertex Method

Return the vertex referenced by the provided object identifier. If no vertex is referenced by that identifier, then return null.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertex GetVertex(  
    Object id  
)
```

VB

```
Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
IVertex^ GetVertex(  
    Object^ id  
)
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

the identifier of the vertex to retrieved from the graph

Return Value

Type: [IVertex](#)

the vertex referenced by the provided identifier or null when no such vertex exists



See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | GetVertices() | Return an iterable to all the vertices in the graph. If this is not possible for the implementation, then an NotSupportedException can be thrown. |
|  | GetVertices(String, Object) | Return an iterable to all the vertices in the graph that have a particular key/value property. If this is not possible for the implementation, then a NotSupportedException can be thrown. The graph implementation should use indexing structures to make this efficient else a full vertex-filter scan is required. |

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetVertices Method

Return an iterable to all the vertices in the graph. If this is not possible for the implementation, then an `NotSupportedException` can be thrown.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IVertex> GetVertices ()
```

VB

```
Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
IEnumerable<IVertex^>^ GetVertices ()
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

an iterable reference to all vertices in the graph

See Also

[IGraph Interface](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.GetVertices Method (String, Object)

Return an iterable to all the vertices in the graph that have a particular key/value property. If this is not possible for the implementation, then a `NotSupportedException` can be thrown. The graph implementation should use indexing structures to make this efficient else a full vertex-filter scan is required.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IVertex> GetVertices (  
    string key,  
    Object value  
)
```

VB

```
Function GetVertices (  
    key As String,  
    value As Object  
) As IEnumerable(Of IVertex)
```

C++

```
IEnumerable<IVertex^>^ GetVertices (  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

the key of vertex

value

Type: [System.Object](#)

the value of the vertex

Return Value

Type: [IEnumerable<IVertex>](#)

an iterable of vertices with provided key and value

VelocityDB Class Library

See Also

[IGraph Interface](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.Query Method

Generate a query object that can be used to fine tune which edges/vertices are retrieved from the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IQuery Query()
```

VB

```
Function Query As IQuery
```

C++

```
IQuery^ Query()
```

F#

```
abstract Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

a graph query object with methods for constraining which data is pulled from the underlying graph

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.RemoveEdge Method

Remove the provided edge from the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
void RemoveEdge (  
    IEdge^ edge  
)
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the edge to remove from the graph

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.RemoveVertex Method

Remove the provided vertex from the graph. Upon removing the vertex, all the edges by which the vertex is connected must be removed as well.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
void RemoveVertex(  
    IVertex^ vertex  
)
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex to remove from the graph

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IGraph.Shutdown Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.IGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Shutdown ()
```

VB

```
Sub Shutdown
```

C++

```
void Shutdown ()
```

F#

```
abstract Shutdown : unit -> unit
```

See Also

[IGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex Interface

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.IIndex"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IIndex
```

VB

```
Public Interface IIndex
```

C++



```
public interface class IIndex
```

F#






```
type IIndex = interface end
```

The **IIndex** type exposes the following members.


Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|--------------------------------------------|
|  | Name | Get the name of the index. |
|  | Type | Get the class that this index is indexing. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Count | Get a count of elements with a particular key/value pair. The semantics are the same as the get method. |
|  | Get | Get all elements that are indexed by the provided key/value. |
|  | Put | Index an element by a key and a value. |
|  | Query | Get all the elements that are indexed by the provided key and specified query object. This is useful for graph implementations that support complex query capabilities. If querying is not supported, simply throw a NotSupportedException. |
|  | Remove | Remove an element indexed by a particular key/value. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

VelocityDB Class Library



See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Index Properties

The [IIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|--------------------------------------------|
|  | Name | Get the name of the index. |
|  | Type | Get the class that this index is indexing. |

See Also

[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Name Property

Get the name of the index.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
string Name { get; }
```

VB

```
ReadOnly Property Name As String  
    Get
```

C++

```
property String^ Name {  
    String^ get ();  
}
```

F#

```
abstract Name : string with get
```

Property Value

Type: [String](#)

the name of the index

See Also

[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Type Property

Get the class that this index is indexing.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Type Type { get; }
```

VB

```
ReadOnly Property Type As Type  
    Get
```

C++

```
property Type^ Type {  
    Type^ get ();  
}
```

F#

```
abstract Type : Type with get
```

Property Value

Type: [Type](#)

the class this index is indexing

See Also






[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)


IIndex.Index Methods

The [IIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Count | Get a count of elements with a particular key/value pair. The semantics are the same as the get method. |
|  | Get | Get all elements that are indexed by the provided key/value. |
|  | Put | Index an element by a key and a value. |
|  | Query | Get all the elements that are indexed by the provided key and specified query object. This is useful for graph implementations that support complex query capabilities. If querying is not supported, simply throw a NotSupportedException. |
|  | Remove | Remove an element indexed by a particular key/value. |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Count Method

Get a count of elements with a particular key/value pair. The semantics are the same as the get method.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
long Count(  
    string key,  
    Object value  
)
```

VB

```
Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
long long Count(  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

denoting the sub-index to search

value

Type: [System.Object](#)

the value to search for

Return Value

Type: [Int64](#)

the collection of elements that meet that criteria

See Also

[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Get Method

Get all elements that are indexed by the provided key/value.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IElement> Get(  
    string key,  
    Object value  
)
```

VB

```
Function Get (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
IEnumerable<IElement^> Get(  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract Get :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

the key of the indexed elements

value

Type: [System.Object](#)

the value of the indexed elements

Return Value

Type: [IEnumerable\(IElement\)](#)

an IEnumerable of elements that have a particular key/value in the index

See Also

[IIndex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Put Method

Index an element by a key and a value.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Put(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
)
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

the key to index the element by

value

Type: [System.Object](#)

the value to index the element by

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the element to index

VelocityDB Class Library

See Also

[Index Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Query Method

Get all the elements that are indexed by the provided key and specified query object. This is useful for graph implementations that support complex query capabilities. If querying is not supported, simply throw a NotSupportedException.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IElement> Query(  
    string key,  
    Object query  
)
```

VB

```
Function Query (  
    key As String,  
    query As Object  
) As IEnumerable(Of IElement)
```

C++

```
IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ query  
)
```

F#

```
abstract Query :  
    key : string *  
    query : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

the key of the indexed elements

query

Type: [System.Object](#)

the query object for the indexed elements' keys

Return Value

Type: [IEnumerable\(IElement\)](#)

an IEnumerable of elements that have a particular key/value in the index that match the query object

VelocityDB Class Library

See Also

[Index Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndex.Remove Method

Remove an element indexed by a particular key/value.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
)
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

the key of the indexed element

value

Type: [System.Object](#)

the value of the indexed element

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the element to remove given the key/value pair

VelocityDB Class Library

See Also

[Index Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndexableGraph Interface

An IIndexableGraph is a graph that supports the manual indexing of its elements. An index is typically some sort of tree structure that allows for the fast lookup of elements by key/value pairs. Indices have an Index object associated with them and allow the user to specify the putting and getting of elements into the index.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IIndexableGraph : IGraph
```

VB

```
Public Interface IIndexableGraph
    Inherits IGraph
```

C++





```
public interface class IIndexableGraph : IGraph
```

F#


```
type IIndexableGraph =
    interface
        interface IGraph
    end
```














The **IIndexableGraph** type exposes the following members.

Methods

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  CreateIndex | Generate an index with a particular name for a particular class. |
|  DropIndex | Remove an index associated with the graph. |
|  GetIndex | Get an index from the graph by its name and index class. An index is unique up to name. |
|  GetIndices | Get all the indices maintained by the graph. |

Extension Methods

| Name | Description |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |





See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)














IIndexableGraph.IIndexableGraph Methods

The [IIndexableGraph](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  CreateIndex | Generate an index with a particular name for a particular class. |
|  DropIndex | Remove an index associated with the graph. |
|  GetIndex | Get an index from the graph by its name and index class. An index is unique up to name. |
|  GetIndices | Get all the indices maintained by the graph. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|
|  | SaveGraphson | (Defined by GraphHelpers.) |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|

See Also

[IIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndexableGraph.CreateIndex Method

Generate an index with a particular name for a particular class.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

the name of the manual index

indexClass

Type: [System.Type](#)

the element class that this index is indexing (can be base class)

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

a collection of parameters for the underlying index implementation

Return Value

Type: [IIndex](#)

the index created

See Also

[IIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndexableGraph.DropIndex Method

Remove an index associated with the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void DropIndex(  
    string indexName  
)
```

VB

```
Sub DropIndex (  
    indexName As String  
)
```

C++

```
void DropIndex(  
    String^ indexName  
)
```

F#

```
abstract DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

the name of the index to drop

See Also

[IIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndexableGraph.GetIndex Method

Get an index from the graph by its name and index class. An index is unique up to name.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
)
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

the name of the index to retrieve

indexClass

Type: [System.Type](#)

the class of the elements being indexed (can be base class)

Return Value

Type: [IIndex](#)

the retrieved index

See Also

[IIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IIndexableGraph.GetIndices Method

Get all the indices maintained by the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IIndex> GetIndices ()
```

VB

```
Function GetIndices As IEnumerable(Of IIndex)
```

C++

```
IEnumerable<IIndex^>^ GetIndices ()
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

the indices associated with the graph

See Also

[IIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IKeyIndexableGraph Interface

A KeyIndexableGraph is a graph that supports basic index functionality around the key/value pairs of the elements of the graph. By creating key indices for a particular property key, that key is indexed on all the elements of the graph. This has ramifications for quick lookups on methods like `getVertices(string, object)` and `getEdges(string, object)`.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IKeyIndexableGraph : IGraph
```

VB

```
Public Interface IKeyIndexableGraph
    Inherits IGraph
```

C++




```
public interface class IKeyIndexableGraph : IGraph
```

F#



```
type IKeyIndexableGraph =
    interface
        interface IGraph
    end
```












The **IKeyIndexableGraph** type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  CreateKeyIndex | Create an automatic indexing structure for indexing provided key for element class. |
|  DropKeyIndex | Remove an automatic indexing structure associated with indexing provided key for element class. |
|  GetIndexedKeys | Return all the index keys associated with a particular element class. |

Extension Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |




See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)














IKeyIndexableGraph.IKeyIndexableGraph Methods

The [IKeyIndexableGraph](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  CreateKeyIndex | Create an automatic indexing structure for indexing provided key for element class. |
|  DropKeyIndex | Remove an automatic indexing structure associated with indexing provided key for element class. |
|  GetIndexedKeys | Return all the index keys associated with a particular element class. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

VelocityDB Class Library

See Also

[IKeyIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IKeyIndexableGraph.CreateKeyIndex Method

Create an automatic indexing structure for indexing provided key for element class.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void CreateKeyIndex(  
    string key,  
    Type elementClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Sub CreateKeyIndex (  
    key As String,  
    elementClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
void CreateKeyIndex(  
    String^ key,  
    Type^ elementClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
abstract CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

key

Type: [System.String](#)

the key to drop the index for

elementClass

Type: [System.Type](#)

the element class that the index is for

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

a collection of parameters for the underlying index implementation

VelocityDB Class Library

See Also

[IKeyIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IKeyIndexableGraph.DropKeyIndex Method

Remove an automatic indexing structure associated with indexing provided key for element class.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void DropKeyIndex(  
    string key,  
    Type elementClass  
)
```

VB

```
Sub DropKeyIndex (  
    key As String,  
    elementClass As Type  
)
```

C++

```
void DropKeyIndex(  
    String^ key,  
    Type^ elementClass  
)
```

F#

```
abstract DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit
```

Parameters

key

Type: [System.String](#)

the key to drop the index for

elementClass

Type: [System.Type](#)

the element class that the index is for

See Also

[IKeyIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IKeyIndexableGraph.GetIndexedKeys Method

Return all the index keys associated with a particular element class.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<string> GetIndexedKeys (  
    Type elementClass  
)
```

VB

```
Function GetIndexedKeys (  
    elementClass As Type  
) As IEnumerable(Of String)
```

C++

```
IEnumerable<String^>^ GetIndexedKeys (  
    Type^ elementClass  
)
```

F#

```
abstract GetIndexedKeys :  
    elementClass : Type -> IEnumerable<string>
```

Parameters

elementClass

Type: [System.Type](#)

the element class that the index is for

Return Value

Type: [IEnumerable\(String\)](#)

keys as a Set

See Also

[IKeyIndexableGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery Interface

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.IQuery"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IQuery
```

VB

```
Public Interface IQuery
```

C++







```
public interface class IQuery
```

F#

```
type IQuery = interface end
```

The **IQuery** type exposes the following members.

Methods

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|  Edges | Execute the query and return the matching edges. |
|  Has(String, Object) | Filter out the edge if it does not have a property with the specified value. |
|  Has(T)(String, Compare, T) | Filter out the edge if it does not have a property with a comparable value. |
|  Interval(T) | Filter out the edge if its property value is not within the provided interval. |
|  Limit | Filter out the edge if the max number of edges to retrieve has already been reached. |
|  Vertices | Execute the query and return the vertices on the other end of the matching edges. |







See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.IQuery Methods

The [IQuery](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|  Edges | Execute the query and return the matching edges. |
|  Has(String, Object) | Filter out the edge if it does not have a property with the specified value. |
|  Has(T)(String, Compare, T) | Filter out the edge if it does not have a property with a comparable value. |
|  Interval(T) | Filter out the edge if its property value is not within the provided interval. |
|  Limit | Filter out the edge if the max number of edges to retrieve has already been reached. |
|  Vertices | Execute the query and return the vertices on the other end of the matching edges. |

See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Edges Method

Execute the query and return the matching edges.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IEdge> Edges ()
```

VB

```
Function Edges As IEnumerable (Of IEdge)
```

C++

```
IEnumerable<IEdge^> Edges ()
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

the unfiltered edges



See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Has Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|------------------------------------------------------------------------------|
|  | Has(String, Object) | Filter out the edge if it does not have a property with the specified value. |
|  | Has(T)(String, Compare, T) | Filter out the edge if it does not have a property with a comparable value. |

See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Has Method (String, Object)

Filter out the edge if it does not have a property with the specified value.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IQuery Has (  
    string key,  
    Object value  
)
```

VB

```
Function Has (  
    key As String,  
    value As Object  
) As IQuery
```

C++

```
IQuery^ Has (  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract Has :  
    key : string *  
    value : Object -> IQuery
```

Parameters

key

Type: [System.String](#)

the key of the property

value

Type: [System.Object](#)

the value to check against

Return Value

Type: [IQuery](#)

the modified query object

See Also

[IQuery Interface](#)

[Has Overload](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Has(T) Method (String, Compare, T)

Filter out the edge if it does not have a property with a comparable value.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IQuery Has<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Function Has(Of T) (
    key As String,
    compare As Compare,
    value As T
) As IQuery
```

C++

```
generic<typename T>
IQuery^ Has(
    String^ key,
    Compare compare,
    T value
)
```

F#

```
abstract Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

the key of the property

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

the comparator to use for comparison

value

Type: **T**

the value to check against

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.IQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Return Value

Type: [IQuery](#)

the modified query object

See Also

[IQuery Interface](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Interval(*T*) Method

Filter out the edge of its property value is not within the provided interval.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IQuery Interval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Function Interval(Of T) (
    key As String,
    startValue As T,
    endValue As T
) As IQuery
```

C++

```
generic<typename T>
IQuery^ Interval(
    String^ key,
    T startValue,
    T endValue
)
```

F#

```
abstract Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

the key of the property

startValue

Type: **T**

the inclusive start value of the interval

endValue

Type: **T**

VelocityDB Class Library

the exclusive end value of the interval

Type Parameters

T

**[Missing <typeparam name="T"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.IQuery.Interval`1(System.String,`0,`0)"]**

Return Value

Type: [IQuery](#)

the modified query object

See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Limit Method

Filter out the edge if the max number of edges to retrieve has already been reached.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IQuery Limit(  
    long max  
)
```

VB

```
Function Limit (  
    max As Long  
) As IQuery
```

C++

```
IQuery^ Limit(  
    long long max  
)
```

F#

```
abstract Limit :  
    max : int64 -> IQuery
```

Parameters

max

Type: [System.Int64](#)

the max number of edges to return

Return Value

Type: [IQuery](#)

the modified query object

See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IQuery.Vertices Method

Execute the query and return the vertices on the other end of the matching edges.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IVertex> Vertices ()
```

VB

```
Function Vertices As IEnumerable(Of IVertex)
```

C++

```
IEnumerable<IVertex^> Vertices ()
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable\(IVertex\)](#)

the unfiltered edge's vertices

See Also

[IQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IThreadedTransactionalGraph Interface

ThreadedTransactionalGraph provides more fine grained control over the transactional context. While TransactionalGraph binds each transaction to the executing thread, ThreadedTransactionalGraph's newTransaction returns a TransactionalGraph that represents its own transactional context independent of the executing thread. Hence, one can have multiple threads operating against a single transaction represented by the returned TransactionalGraph object. This is useful for parallelizing graph algorithms. Note, that one needs to call TransactionalGraph.Commit() or TransactionalGraph.Rollback() to close the transactions returned

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IThreadedTransactionalGraph : ITransactionalGraph,
    IGraph
```

VB

```
Public Interface IThreadedTransactionalGraph
    Inherits ITransactionalGraph, IGraph
```

C++


```
public interface class IThreadedTransactionalGraph : ITransactionalGraph,
    IGraph
```

F#


```
type IThreadedTransactionalGraph =
    interface
        interface ITransactionalGraph
        interface IGraph
    end
```













The **IThreadedTransactionalGraph** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|----------------------------------------------------------------------------------------------------------------|
|  | NewTransaction | Returns a TransactionalGraph that represents a transactional context independent of the executing transaction. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |


See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)














IThreadedTransactionalGraph.IThreadedTransactionalGraph Methods

The [IThreadedTransactionalGraph](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|  NewTransaction | Returns a TransactionalGraph that represents a transactional context independent of the executing transaction. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[IThreadedTransactionalGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IThreadedTransactionalGraph.NewTransaction Method

Returns a TransactionalGraph that represents a transactional context independent of the executing transaction.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
ITransactionalGraph NewTransaction()
```

VB

```
Function NewTransaction As ITransactionalGraph
```

C++

```
ITransactionalGraph^ NewTransaction()
```

F#

```
abstract NewTransaction : unit -> ITransactionalGraph
```

Return Value

Type: [ITransactionalGraph](#)

A transactional context. Invoking TransactionalGraph.shutdown() successfully commits the transaction.

See Also

[IThreadedTransactionalGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

ITransactionalGraph Interface

A transactional graph supports the notion of transactions. A transaction scopes a logically coherent operation composed of multiple read and write operations that either occurs at once or not at all. The exact notion of a transaction and its isolational guarantees depend on the implementing graph database. A transaction scopes a coherent and complete operations. Any element references created during a transaction should not be accessed outside its scope (i.e. after the transaction is committed or rolled back). Accessing such references outside the transactional context they were created in may lead to exceptions. If such access is necessary, the transactional context should be extended. By default, the first operation on a TransactionalGraph will start a transaction automatically.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface ITransactionalGraph : IGraph
```

VB

```
Public Interface ITransactionalGraph
    Inherits IGraph
```

C++



```
public interface class ITransactionalGraph : IGraph
```

F#


```
type ITransactionalGraph =
    interface
        interface IGraph
    end
```













The **ITransactionalGraph** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------|
|  | Commit | Stop the current transaction and successfully apply mutations to the graph. |
|  | Rollback | Stop the current transaction and drop any mutations applied since the last transaction. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |



See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)














ITransactionalGraph.ITransactionalGraph Methods

The [ITransactionalGraph](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
|  Commit | Stop the current transaction and successfully apply mutations to the graph. |
|  Rollback | Stop the current transaction and drop any mutations applied since the last transaction. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[ITransactionalGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

ITransactionalGraph.Commit Method

Stop the current transaction and successfully apply mutations to the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Commit ()
```

VB

```
Sub Commit
```

C++

```
void Commit ()
```

F#

```
abstract Commit : unit -> unit
```

See Also

[ITransactionalGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

ITransactionalGraph.Rollback Method

Stop the current transaction and drop any mutations applied since the last transaction.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Rollback()
```

VB

```
Sub Rollback
```

C++

```
void Rollback()
```

F#

```
abstract Rollback : unit -> unit
```

See Also

[ITransactionalGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertex Interface

A vertex maintains pointers to both a set of incoming and outgoing edges. The outgoing edges are those edges for which the vertex is the tail. The incoming edges are those edges for which the vertex is the head. Diagrammatically, ---inEdges---> vertex ---outEdges--->.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IVertex : IElement,
    IDictionary<string, Object>, ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection
```

VB

```
Public Interface IVertex
    Inherits IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++


```
public interface class IVertex : IElement,
    IDictionary<String^, Object^>, ICollection<KeyValuePair<String^,
Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>, IEnumerable,
    IDictionary, ICollection
```




F#

```
type IVertex =
    interface
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
        interface ICollection
    end
```












The **IVertex** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|--------------------------------------------------------------------------------------------|
|  | AddEdge | Add a new outgoing edge from this vertex to the parameter vertex with provided edge label. |

| | |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GetEdges | Return the edges incident to the vertex according to the provided direction and edge labels. |
|  GetVertices | Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge). |
|  Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved that are incident/adjacent to this vertex. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |

VelocityDB Class Library





See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)










IVertex Methods



The [IVertex](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add a new outgoing edge from this vertex to the parameter vertex with provided edge label. |
|  GetEdges | Return the edges incident to the vertex according to the provided direction and edge labels. |
|  GetVertices | Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge). |
|  Query | Generate a query object that can be used to fine tune which edges/vertices are retrieved that are incident/adjacent to this vertex. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be |

| | | |
|-----------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  | VertexString | (Defined by StringFactory.) |

See Also

[IVertex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertex.AddEdge Method

Add a new outgoing edge from this vertex to the parameter vertex with provided edge label.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEdge AddEdge(  
    Object id,  
    string label,  
    IVertex inVertex  
)
```

VB

```
Function AddEdge (  
    id As Object,  
    label As String,  
    inVertex As IVertex  
) As IEdge
```

C++

```
IEdge^ AddEdge(  
    Object^ id,  
    String^ label,  
    IVertex^ inVertex  
)
```

F#

```
abstract AddEdge :  
    id : Object *  
    label : string *  
    inVertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

the id of the edge

label

Type: [System.String](#)

the label of the edge

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.IVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

the newly created edge

See Also

[IVertex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertex.GetEdges Method

Return the edges incident to the vertex according to the provided direction and edge labels.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IEdge> GetEdges (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Function GetEdges (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IEdge)
```

C++

```
IEnumerable<IEdge^>^ GetEdges (  
    Direction direction,  
    ... array<String^>^ labels  
)
```

F#

```
abstract GetEdges :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

the direction of the edges to retrieve

labels

Type: [System.String](#)[]

the labels of the edges to retrieve

Return Value

Type: [IEnumerable\(IEdge\)](#)

an IEnumerable of incident edges

See Also

[IVertex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertex.GetVertices Method

Return the vertices adjacent to the vertex according to the provided direction and edge labels. This method does not remove duplicate vertices (i.e. those vertices that are connected by more than one edge).

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<IVertex> GetVertices (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Function GetVertices (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IVertex)
```

C++

```
IEnumerable<IVertex^>^ GetVertices (  
    Direction direction,  
    ... array<String^>^ labels  
)
```

F#

```
abstract GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

the direction of the edges of the adjacent vertices

labels

Type: [System.String\[\]](#)

the labels of the edges of the adjacent vertices

Return Value

Type: [IEnumerable<IVertex>](#)

an IEnumerable of adjacent vertices

VelocityDB Class Library

See Also

[IVertex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertex.Query Method

Generate a query object that can be used to fine tune which edges/vertices are retrieved that are incident/adjacent to this vertex.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertexQuery Query()
```

VB

```
Function Query As IVertexQuery
```

C++

```
IVertexQuery^ Query()
```

F#

```
abstract Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

a vertex query object with methods for constraining which data is pulled from the underlying graph

See Also

[IVertex Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery Interface

A VertexQuery object defines a collection of filters and modifiers that are used to intelligently select edges from a vertex.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IVertexQuery : IQuery
```

VB

```
Public Interface IVertexQuery
    Inherits IQuery
```

C++





```
public interface class IVertexQuery : IQuery
```

F#

```
type IVertexQuery =
    interface
        interface IQuery
    end
```

The **IVertexQuery** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------|
|  | Count | Execute the query and return the number of edges that are unfiltered. |
|  | Direction | The direction of the edges to retrieve. |
|  | Labels | Filter out the edge if its label is not in set of provided labels. |
|  | VertexIds | Return the raw ids of the vertices on the other end of the edges. |





See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery.IVertexQuery Methods

The [IVertexQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-----------------------------------------------------------------------|
|  | Count | Execute the query and return the number of edges that are unfiltered. |
|  | Direction | The direction of the edges to retrieve. |
|  | Labels | Filter out the edge if its label is not in set of provided labels. |
|  | VertexIds | Return the raw ids of the vertices on the other end of the edges. |

See Also

[IVertexQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery.Count Method

Execute the query and return the number of edges that are unfiltered.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
long Count ()
```

VB

```
Function Count As Long
```

C++

```
long long Count ()
```

F#

```
abstract Count : unit -> int64
```

Return Value

Type: [Int64](#)

the number of unfiltered edges

See Also

[IVertexQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery.Direction Method

The direction of the edges to retrieve.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertexQuery Direction(  
    Direction direction  
)
```

VB

```
Function Direction (  
    direction As Direction  
) As IVertexQuery
```

C++

```
IVertexQuery^ Direction(  
    Direction direction  
)
```

F#

```
abstract Direction :  
    direction : Direction -> IVertexQuery
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

whether to retrieve the incoming, outgoing, or both directions

Return Value

Type: [IVertexQuery](#)

the modified query object

See Also

[IVertexQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery.Labels Method

Filter out the edge if its label is not in set of provided labels.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertexQuery Labels(  
    params string[] labels  
)
```

VB

```
Function Labels (  
    ParamArray labels As String()  
) As IVertexQuery
```

C++

```
IVertexQuery^ Labels(  
    ... array<String^>^ labels  
)
```

F#

```
abstract Labels :  
    labels : string[] -> IVertexQuery
```

Parameters

labels

Type: [System.String\[\]](#)

the labels to check against

Return Value

Type: [IVertexQuery](#)

the modified query object

See Also

[IVertexQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

IVertexQuery.VertexIds Method

Return the raw ids of the vertices on the other end of the edges.

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEnumerable<Object> VertexIds ()
```

VB

```
Function VertexIds As IEnumerable (Of Object)
```

C++

```
IEnumerable<Object^>^ VertexIds ()
```

F#

```
abstract VertexIds : unit -> IEnumerable<Object>
```

Return Value

Type: [IEnumerable\(Object\)](#)

the raw ids of the vertices on the other end of the edges

See Also

[IVertexQuery Interface](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Parameter"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Parameter

[VelocityGraph.Frontenac.Blueprints.Parameter\(TK, TV\)](#)

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Parameter
```

VB

```
Public Class Parameter
```

C++


```
public ref class Parameter
```

F#



```
type Parameter = class end
```

The **Parameter** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------|
|  | Parameter | Initializes a new instance of the Parameter class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|-------------|
|  | Key | |
|  | Value | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |

| | | |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | ToString | (Overrides Object.ToString() .) |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter Constructor

Initializes a new instance of the [Parameter](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Parameter(  
    Object key,  
    Object value  
)
```

VB

```
Public Sub New (  
    key As Object,  
    value As Object  
)
```

C++

```
public:  
Parameter(  
    Object^ key,  
    Object^ value  
)
```

F#

```
new :  
    key : Object *  
    value : Object -> Parameter
```

Parameters

key

Type: [System.Object](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Parameter.#ctor(System.Object,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Parameter.#ctor(System.Object,System.Object)"]

See Also



[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.Parameter Properties

The [Parameter](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Key | |
|  | Value | |

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.Key Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Parameter.Key"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object Key { get; }
```

VB

```
Public ReadOnly Property Key As Object  
    Get
```

C++

```
public:  
property Object^ Key {  
    Object^ get ();  
}
```

F#

```
member Key : Object with get
```

Property Value

Type: [Object](#)

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.Value Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Parameter.Value"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object Value { get; set; }
```

VB

```
Public Property Value As Object  
    Get  
    Set
```

C++

```
public:  
property Object^ Value {  
    Object^ get ();  
    void set (Object^ value);  
}
```

F#

```
member Value : Object with get, set
```

Property Value

Type: [Object](#)

See Also




[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.Parameter Methods

The [Parameter](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object).) |
|  | GetHashCode | (Overrides Object.GetHashCode().) |
|  | ToString | (Overrides Object.ToString().) |

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.Equals Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.Equals(System.Object)"]

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.GetHashCode Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.GetHashCode"]

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter.ToString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter.ToString"]

See Also

[Parameter Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter(TK, TV) Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Parameter`2"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Parameter](#)

VelocityGraph.Frontenac.Blueprints.Parameter(TK, TV)

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class Parameter<TK, TV> : Parameter
```

VB

```
Public Class Parameter(Of TK, TV)  
    Inherits Parameter
```

C++

```
generic<typename TK, typename TV>  
public ref class Parameter : public Parameter
```

F#

```
type Parameter<'TK, 'TV> =  
    class  
        inherit Parameter  
    end
```

Type Parameters

TK


[Missing <typeparam name="TK"/> documentation for "T:VelocityGraph.Frontenac.Blueprints.Parameter`2"]

TV



[Missing <typeparam name="TV"/> documentation for "T:VelocityGraph.Frontenac.Blueprints.Parameter`2"]

The Parameter(TK, TV) type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-----------------------------------------------------------|
|  | Parameter(TK, TV) | Initializes a new instance of the Parameter(TK, TV) class |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Key | |
|  | Value | |

See Also

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter(TK, TV) Constructor

Initializes a new instance of the [Parameter\(TK, TV\)](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Parameter(  
    TK key,  
    TV value  
)
```

VB

```
Public Sub New (  
    key As TK,  
    value As TV  
)
```

C++

```
public:  
Parameter(  
    TK key,  
    TV value  
)
```

F#

```
new :  
    key : 'TK *  
    value : 'TV -> Parameter
```

Parameters

key

Type: *TK*

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter`2.#ctor(`0,`1)"]

value

Type: *TV*

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Parameter`2.#ctor(`0,`1)"]

See Also



[Parameter\(TK, TV\)Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter(TK, TV).Parameter(TK, TV) Properties

The [Parameter\(TK, TV\)](#) generic type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Key | |
|  | Value | |

See Also

[Parameter\(TK, TV\)Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter(TK, TV).Key Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Parameter`2.Key"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TK Key { get; }
```

VB

```
Public ReadOnly Property Key As TK  
    Get
```

C++

```
public:  
property TK Key {  
    TK get ();  
}
```

F#

```
member Key : 'TK with get
```

Property Value

Type: *TK*

See Also

[Parameter\(TK, TV\)Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

Parameter(TK, TV).Value Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Parameter`2.Value"]

Namespace: [VelocityGraph.Frontenac.Blueprints](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TV Value { get; set; }
```

VB

```
Public Property Value As TV  
    Get  
    Set
```

C++

```
public:  
property TV Value {  
    TV get ();  
    void set (TV value);  
}
```

F#

```
member Value : 'TV with get, set
```

Property Value

Type: *TV*

See Also




[Parameter\(TK, TV\)Class](#)

[VelocityGraph.Frontenac.Blueprints Namespace](#)

VelocityGraph.Frontenac.Blueprints.Contracts Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Contracts"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------|-------------|
|  | EdgeContract | |
|  | ElementContract | |
|  | GraphContract | |
|  | IndexableGraphContract | |
|  | IndexContract | |
|  | KeyIndexableGraphContract | |
|  | QueryContract | |
|  | ThreadedTransactionalGraphContract | |
|  | VertexContract | |
|  | VertexQueryContract | |

EdgeContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Contracts.EdgeContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.EdgeContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class EdgeContract
```

VB

```
Public NotInheritable Class EdgeContract
```

C++

```
public ref class EdgeContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type EdgeContract = class end
```

The **EdgeContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | ValidateGetVertex | |


See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

EdgeContract.EdgeContract Methods

The [EdgeContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | ValidateGetVertex | |

See Also

[EdgeContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

EdgeContract.ValidateGetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.EdgeContract.ValidateGetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetVertex(  
    Direction direction  
)
```

VB

```
Public Shared Sub ValidateGetVertex (  
    direction As Direction  
)
```

C++

```
public:  
static void ValidateGetVertex(  
    Direction direction  
)
```

F#

```
static member ValidateGetVertex :  
    direction : Direction -> unit
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.EdgeContract.ValidateGetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

See Also

[EdgeContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ElementContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class ElementContract
```

VB

```
Public NotInheritable Class ElementContract
```

C++




```
public ref class ElementContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type ElementContract = class end
```

The **ElementContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------|-------------|
|  | ValidateGetProperty | |
|  | ValidateRemoveProperty | |
|  | ValidateSetProperty | |




See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ElementContract.ElementContract Methods

The [ElementContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|-------------|
|  | ValidateGetProperty | |
|  | ValidateRemoveProperty | |
|  | ValidateSetProperty | |

See Also

[ElementContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ElementContract.ValidateGetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateGetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetProperty(  
    string key  
)
```

VB

```
Public Shared Sub ValidateGetProperty (  
    key As String  
)
```

C++

```
public:  
static void ValidateGetProperty(  
    String^ key  
)
```

F#

```
static member ValidateGetProperty :  
    key : string -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateGetProperty(System.String)"]

See Also

[ElementContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ElementContract.ValidateRemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateRemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateRemoveProperty(  
    string key  
)
```

VB

```
Public Shared Sub ValidateRemoveProperty (  
    key As String  
)
```

C++

```
public:  
static void ValidateRemoveProperty(  
    String^ key  
)
```

F#

```
static member ValidateRemoveProperty :  
    key : string -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateRemoveProperty(System.String)"]

See Also

[ElementContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ElementContract.ValidateSetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateSetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateSetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Shared Sub ValidateSetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
static void ValidateSetProperty(  
    String^ key,  
    Object^ value  
)
```

F#

```
static member ValidateSetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateSetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ElementContract.ValidateSetProperty(System.String,System.Object)"]

VelocityDB Class Library

See Also

[ElementContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphContract
```

VB

```
Public NotInheritable Class GraphContract
```

C++








```
public ref class GraphContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]
[<SealedAttribute>]
type GraphContract = class end
```

The **GraphContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | ValidateAddEdge | |
|  | ValidateGetEdge | |
|  | ValidateGetEdges | |
|  | ValidateGetVertex | |
|  | ValidateGetVertices | |
|  | ValidateRemoveEdge | |
|  | ValidateRemoveVertex | |








See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.GraphContract Methods

The [GraphContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | ValidateAddEdge | |
|  | ValidateGetEdge | |
|  | ValidateGetEdges | |
|  | ValidateGetVertex | |
|  | ValidateGetVertices | |
|  | ValidateRemoveEdge | |
|  | ValidateRemoveVertex | |

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateAddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateAddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateAddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Shared Sub ValidateAddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
)
```

C++

```
public:
static void ValidateAddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
)
```

F#

```
static member ValidateAddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> unit
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateAddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateAddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateAddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateAddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateGetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetEdge (  
    Object id  
)
```

VB

```
Public Shared Sub ValidateGetEdge (  
    id As Object  
)
```

C++

```
public:  
static void ValidateGetEdge (  
    Object^ id  
)
```

F#

```
static member ValidateGetEdge :  
    id : Object -> unit
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetEdge(System.Object)"]

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateGetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Shared Sub ValidateGetEdges (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
static void ValidateGetEdges (  
    String^ key,  
    Object^ value  
)
```

F#

```
static member ValidateGetEdges :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetEdges(System.String,System.Object)"]

VelocityDB Class Library

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateGetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetVertex(  
    Object id  
)
```

VB

```
Public Shared Sub ValidateGetVertex (  
    id As Object  
)
```

C++

```
public:  
static void ValidateGetVertex(  
    Object^ id  
)
```

F#

```
static member ValidateGetVertex :  
    id : Object -> unit
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetVertex(System.Object)"]

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateGetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetVertices (  
    string key,  
    Object value  
)
```

VB

```
Public Shared Sub ValidateGetVertices (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
static void ValidateGetVertices (  
    String^ key,  
    Object^ value  
)
```

F#

```
static member ValidateGetVertices :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateGetVertices(System.String,System.Object)"]

VelocityDB Class Library

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateRemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateRemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateRemoveEdge (  
    IEdge edge  
)
```

VB

```
Public Shared Sub ValidateRemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
static void ValidateRemoveEdge (  
    IEdge^ edge  
)
```

F#

```
static member ValidateRemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateRemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

GraphContract.ValidateRemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateRemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateRemoveVertex (  
    IVertex vertex  
)
```

VB

```
Public Shared Sub ValidateRemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
static void ValidateRemoveVertex (  
    IVertex^ vertex  
)
```

F#

```
static member ValidateRemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.GraphContract.ValidateRemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also

[GraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexableGraphContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class IndexableGraphContract
```

VB

```
Public NotInheritable Class IndexableGraphContract
```

C++




```
public ref class IndexableGraphContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type IndexableGraphContract = class end
```

The **IndexableGraphContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|-------------|
|  | ValidateCreateIndex | |
|  | ValidateDropIndex | |
|  | ValidateGetIndex | |




See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexableGraphContract.IndexableGraphContract Methods

The [IndexableGraphContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------|-------------|
|  | ValidateCreateIndex | |
|  | ValidateDropIndex | |
|  | ValidateGetIndex | |

See Also

[IndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexableGraphContract.ValidateCreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateCreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateCreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Shared Sub ValidateCreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
public:  
static void ValidateCreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
static member ValidateCreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateCreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateCreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateCreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

See Also

[IndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexableGraphContract.ValidateDropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateDropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateDropIndex(  
    string indexName  
)
```

VB

```
Public Shared Sub ValidateDropIndex (  
    indexName As String  
)
```

C++

```
public:  
static void ValidateDropIndex(  
    String^ indexName  
)
```

F#

```
static member ValidateDropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateDropIndex(System.String)"]

See Also

[IndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexableGraphContract.ValidateGetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateGetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Shared Sub ValidateGetIndex (  
    indexName As String,  
    indexClass As Type  
)
```

C++

```
public:  
static void ValidateGetIndex(  
    String^ indexName,  
    Type^ indexClass  
)
```

F#

```
static member ValidateGetIndex :  
    indexName : string *  
    indexClass : Type -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateGetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexableGraphContract.ValidateGetIndex(System.String,System.Type)"]

VelocityDB Class Library

See Also

[IndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class IndexContract
```

VB

```
Public NotInheritable Class IndexContract
```

C++






```
public ref class IndexContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type IndexContract = class end
```

The **IndexContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | ValidateCount | |
|  | ValidateGet | |
|  | ValidatePut | |
|  | ValidateQuery | |
|  | ValidateRemove | |






See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.IndexContract Methods

The [IndexContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | ValidateCount | |
|  | ValidateGet | |
|  | ValidatePut | |
|  | ValidateQuery | |
|  | ValidateRemove | |

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.ValidateCount Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateCount(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateCount (
    string key,
    Object value
)
```

VB

```
Public Shared Sub ValidateCount (
    key As String,
    value As Object
)
```

C++

```
public:
static void ValidateCount (
    String^ key,
    Object^ value
)
```

F#

```
static member ValidateCount :
    key : string *
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateCount(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateCount(System.String,System.Object)"]

VelocityDB Class Library

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.ValidateGet Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateGet(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGet (
    string key,
    Object value
)
```

VB

```
Public Shared Sub ValidateGet (
    key As String,
    value As Object
)
```

C++

```
public:
static void ValidateGet (
    String^ key,
    Object^ value
)
```

F#

```
static member ValidateGet :
    key : string *
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateGet(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateGet(System.String,System.Object)"]

VelocityDB Class Library

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.ValidatePut Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidatePut(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidatePut (
    string key,
    Object value,
    IElement element
)
```

VB

```
Public Shared Sub ValidatePut (
    key As String,
    value As Object,
    element As IElement
)
```

C++

```
public:
static void ValidatePut (
    String^ key,
    Object^ value,
    IElement^ element
)
```

F#

```
static member ValidatePut :
    key : string *
    value : Object *
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidatePut(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidatePut(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidatePut(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.ValidateQuery Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateQuery(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateQuery(  
    string key,  
    Object query  
)
```

VB

```
Public Shared Sub ValidateQuery (  
    key As String,  
    query As Object  
)
```

C++

```
public:  
static void ValidateQuery(  
    String^ key,  
    Object^ query  
)
```

F#

```
static member ValidateQuery :  
    key : string *  
    query : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateQuery(System.String,System.Object)"]

query

Type: [System.Object](#)

[Missing <param name="query"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateQuery(System.String,System.Object)"]

VelocityDB Class Library

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

IndexContract.ValidateRemove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateRemove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateRemove (
    string key,
    Object value,
    IElement element
)
```

VB

```
Public Shared Sub ValidateRemove (
    key As String,
    value As Object,
    element As IElement
)
```

C++

```
public:
static void ValidateRemove (
    String^ key,
    Object^ value,
    IElement^ element
)
```

F#

```
static member ValidateRemove :
    key : string *
    value : Object *
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateRemove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateRemove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.IndexContract.ValidateRemove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

See Also

[IndexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

KeyIndexableGraphContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class KeyIndexableGraphContract
```

VB

```
Public NotInheritable Class KeyIndexableGraphContract
```

C++




```
public ref class KeyIndexableGraphContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type KeyIndexableGraphContract = class end
```

The **KeyIndexableGraphContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------|-------------|
|  | ValidateCreateKeyIndex | |
|  | ValidateDropKeyIndex | |
|  | ValidateGetIndexedKeys | |




See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

KeyIndexableGraphContract.KeyIndexableGraphContract Methods

The [KeyIndexableGraphContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------|-------------|
|  | ValidateCreateKeyIndex | |
|  | ValidateDropKeyIndex | |
|  | ValidateGetIndexedKeys | |

See Also

[KeyIndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

KeyIndexableGraphContract.ValidateCreateKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateCreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateCreateKeyIndex(  
    string key,  
    Type elementClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Shared Sub ValidateCreateKeyIndex (  
    key As String,  
    elementClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
public:  
static void ValidateCreateKeyIndex(  
    String^ key,  
    Type^ elementClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
static member ValidateCreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateCreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateCreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])]"

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateCreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])]"

See Also

[KeyIndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

KeyIndexableGraphContract.ValidateDropKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateDropKeyIndex (System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateDropKeyIndex (  
    string key,  
    Type elementClass  
)
```

VB

```
Public Shared Sub ValidateDropKeyIndex (  
    key As String,  
    elementClass As Type  
)
```

C++

```
public:  
static void ValidateDropKeyIndex (  
    String^ key,  
    Type^ elementClass  
)
```

F#

```
static member ValidateDropKeyIndex :  
    key : string *  
    elementClass : Type -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateDropKeyIndex (System.String,System.Type)"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateDropKeyIndex (System.String,System.Type)"]

VelocityDB Class Library

See Also

[KeyIndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

KeyIndexableGraphContract.ValidateGetIndexedKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateGetIndexedKeys(System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetIndexedKeys (  
    Type elementClass  
)
```

VB

```
Public Shared Sub ValidateGetIndexedKeys (  
    elementClass As Type  
)
```

C++

```
public:  
static void ValidateGetIndexedKeys (  
    Type^ elementClass  
)
```

F#

```
static member ValidateGetIndexedKeys :  
    elementClass : Type -> unit
```

Parameters

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.KeyIndexableGraphContract.ValidateGetIndexedKeys(System.Type)"]

See Also

[KeyIndexableGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class QueryContract
```

VB

```
Public NotInheritable Class QueryContract
```

C++





```
public ref class QueryContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type QueryContract = class end
```

The **QueryContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------|-------------|
|  | ValidateHas(String, Object) | |
|  | ValidateHas(T)(String, Compare, T) | |
|  | ValidateInterval(T) | |
|  | ValidateLimit | |





See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.QueryContract Methods

The [QueryContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------|-------------|
|  | ValidateHas(String, Object) | |
|  | ValidateHas(T)(String, Compare, T) | |
|  | ValidateInterval(T) | |
|  | ValidateLimit | |


See Also

[QueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.ValidateHas Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------|-------------|
|  | ValidateHas(String, Object) | |
|  | ValidateHas(T)(String, Compare, T) | |

See Also

[QueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.ValidateHas Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateHas (  
    string key,  
    Object value  
)
```

VB

```
Public Shared Sub ValidateHas (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
static void ValidateHas (  
    String^ key,  
    Object^ value  
)
```

F#

```
static member ValidateHas :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas(System.String,System.Object)"]

VelocityDB Class Library

See Also

[QueryContract Class](#)

[ValidateHas Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.ValidateHas(T) Method (String, Compare, T)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateHas<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Public Shared Sub ValidateHas(Of T) (
    key As String,
    compare As Compare,
    value As T
)
```

C++

```
public:
    generic<typename T>
    static void ValidateHas(
        String^ key,
        Compare compare,
        T value
    )
```

F#

```
static member ValidateHas :
    key : string *
    compare : Compare *
    value : 'T -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

value

Type: *T*

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateHas`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

See Also

[QueryContract Class](#)

[ValidateHas Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.ValidateInterval(T) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateInterval`1(System.String,`0,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateInterval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Public Shared Sub ValidateInterval(Of T) (
    key As String,
    startValue As T,
    endValue As T
)
```

C++

```
public:
    generic<typename T>
    static void ValidateInterval(
        String^ key,
        T startValue,
        T endValue
    )
```

F#

```
static member ValidateInterval :
    key : string *
    startValue : 'T *
    endValue : 'T -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateInterval`1(System.String,`0,`0)"]

startValue

Type: *T*

[Missing <param name="startValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateInterval`1(System.String,`0,`0)"]

endValue

Type: *T*

[Missing <param name="endValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateInterval`1(System.String,`0,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateInterval`1(System.String,`0,`0)"]

See Also

[QueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

QueryContract.ValidateLimit Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateLimit(System.Int64)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateLimit(  
    long max  
)
```

VB

```
Public Shared Sub ValidateLimit (  
    max As Long  
)
```

C++

```
public:  
static void ValidateLimit(  
    long long max  
)
```

F#

```
static member ValidateLimit :  
    max : int64 -> unit
```

Parameters

max

Type: [System.Int64](#)

[Missing <param name="max"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.QueryContract.ValidateLimit(System.Int64)"]

See Also

[QueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class ThreadedTransactionalGraphContract :
    IThreadedTransactionalGraph,
    ITransactionalGraph, IGraph
```

VB

```
Public MustInherit Class ThreadedTransactionalGraphContract
    Implements IThreadedTransactionalGraph, ITransactionalGraph, IGraph
```

C++

```
public ref class ThreadedTransactionalGraphContract abstract :
    IThreadedTransactionalGraph,
    ITransactionalGraph, IGraph
```

F#
















```
[<AbstractClassAttribute>]
type ThreadedTransactionalGraphContract =
    class
        interface IThreadedTransactionalGraph
        interface ITransactionalGraph
        interface IGraph
    end
```

The **ThreadedTransactionalGraphContract** type exposes the following members.










Properties





| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | NewTransaction | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | |
|  | Shutdown | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |


See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.ThreadedTransactionalGraphContract Properties

The [ThreadedTransactionalGraphContract](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract Features Features { get; }
```

VB

```
Public MustOverride ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () abstract;  
}
```

F#

```
abstract Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also
















[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)









ThreadedTransactionalGraphContract.ThreadedTransactionalGraphContract Methods






The [ThreadedTransactionalGraphContract](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | NewTransaction | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | |
|  | Shutdown | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public MustOverride Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) abstract
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(Sys

tem.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IVertex AddVertex(  
    Object id  
)
```

VB

```
Public MustOverride Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) abstract
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.AddVertex(System.Object)"]

Implements

[IGraph.AddVertex\(Object\)](#)

VelocityDB Class Library

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.Commit Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Commit"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Commit ()
```

VB

```
Public MustOverride Sub Commit
```

C++

```
public:  
virtual void Commit () abstract
```

F#

```
abstract Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEdge GetEdge (  
    Object id  
)
```

VB

```
Public MustOverride Function GetEdge (  
    id As Object  
) As IEdge
```

C++

```
public:  
virtual IEdge^ GetEdge (  
    Object^ id  
) abstract
```

F#

```
abstract GetEdge :  
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdge(System.Object)"]

Implements

[IGraph.GetEdge\(Object\)](#)

VelocityDB Class Library

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IEdge> GetEdges ()
```

VB

```
Public MustOverride Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () abstract
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public MustOverride Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) abstract
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges(System.String,System.Object)"]

VelocityDB Class Library

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IVertex GetVertex(  
    Object id  
)
```

VB

```
Public MustOverride Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) abstract
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertex(System.Object)"]

Implements

[IGraph.GetVertex\(Object\)](#)

VelocityDB Class Library

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices"
]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IVertex> GetVertices ()
```

VB

```
Public MustOverride Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () abstract
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices"
]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.GetVertices Method (String, Object)

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public MustOverride Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
) abstract
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.NewTransaction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.NewTransaction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ITransactionalGraph NewTransaction()
```

VB

```
Public Function NewTransaction As ITransactionalGraph
```

C++

```
public:  
virtual ITransactionalGraph^ NewTransaction() sealed
```

F#

```
abstract NewTransaction : unit -> ITransactionalGraph  
override NewTransaction : unit -> ITransactionalGraph
```

Return Value

Type: [ITransactionalGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.NewTransaction"]

Implements

[IThreadedTransactionalGraph.NewTransaction\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IQuery Query()
```

VB

```
Public MustOverride Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() abstract
```

F#

```
abstract Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.RemoveEdge (VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Public MustOverride Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void RemoveEdge (  
    IEdge^ edge  
) abstract
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.RemoveEdge (VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public MustOverride Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) abstract
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.Rollback Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Rollback"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Rollback()
```

VB

```
Public MustOverride Sub Rollback
```

C++

```
public:  
virtual void Rollback() abstract
```

F#

```
abstract Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

ThreadedTransactionalGraphContract.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.ThreadedTransactionalGraphContract.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract void Shutdown ()
```

VB

```
Public MustOverride Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () abstract
```

F#

```
abstract Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[ThreadedTransactionalGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class VertexContract
```

VB

```
Public NotInheritable Class VertexContract
```

C++

```
public ref class VertexContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type VertexContract = class end
```

The **VertexContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|-------------|
|  | ValidateAddEdge | |

See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexContract.VertexContract Methods

The [VertexContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-------------|
|  | ValidateAddEdge | |

See Also

[VertexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexContract.ValidateAddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract.ValidateAddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateAddEdge (
    Object id,
    string label,
    IVertex inVertex
)
```

VB

```
Public Shared Sub ValidateAddEdge (
    id As Object,
    label As String,
    inVertex As IVertex
)
```

C++

```
public:
static void ValidateAddEdge (
    Object^ id,
    String^ label,
    IVertex^ inVertex
)
```

F#

```
static member ValidateAddEdge :
    id : Object *
    label : string *
    inVertex : IVertex -> unit
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract.ValidateAddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract.ValidateAddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexContract.ValidateAddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also

[VertexContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class VertexQueryContract : IVertexQuery,
    IQuery
```

VB

```
Public MustInherit Class VertexQueryContract
    Implements IVertexQuery, IQuery
```

C++







```
public ref class VertexQueryContract abstract : IVertexQuery,
    IQuery
```

F#





```
[<AbstractClassAttribute>]
type VertexQueryContract =
    class
        interface IVertexQuery
        interface IQuery
    end
```

The **VertexQueryContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Count | |
|  | Direction | |
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------|--|
|  | Labels | |
|  | Limit | |
|  | VertexIds | |
|  | Vertices | |











See Also

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.VertexQueryContract Methods

The [VertexQueryContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Count | |
|  | Direction | |
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Labels | |
|  | Limit | |
|  | VertexIds | |
|  | Vertices | |

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Count Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Count"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count ()
```

VB

```
Public Function Count As Long
```

C++

```
public:  
virtual long long Count () sealed
```

F#

```
abstract Count : unit -> int64  
override Count : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Count"]

Implements

[IVertexQuery.Count\(\)](#)

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Direction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Direction(  
    Direction direction  
)
```

VB

```
Public Function Direction (  
    direction As Direction  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Direction(  
    Direction direction  
) sealed
```

F#

```
abstract Direction :  
    direction : Direction -> IVertexQuery  
override Direction :  
    direction : Direction -> IVertexQuery
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IVertexQuery.Direction\(Direction\)](#)

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Edges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IEdge> Edges ()
```

VB

```
Public MustOverride Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ Edges () abstract
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Edges"]

Implements

[IQuery.Edges\(\)](#)

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Has Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Has Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IQuery Has (  
    string key,  
    Object value  
)
```

VB

```
Public MustOverride Function Has (  
    key As String,  
    value As Object  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Has(  
    String^ key,  
    Object^ value  
) abstract
```

F#

```
abstract Has :  
    key : string *  
    value : Object -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has(System.String,System.Object)"]

VelocityDB Class Library

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has(System.String,System.Object)"]

Implements

[IQuery.Has\(String, Object\)](#)

See Also

[VertexQueryContract Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Has(*T*) Method (String, Compare, *T*)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IQuery Has<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Public MustOverride Function Has(Of T) (
    key As String,
    compare As Compare,
    value As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Has(
    String^ key,
    Compare compare,
    T value
) abstract
```

F#

```
abstract Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

value

Type: *T*

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Implements

[IQuery.Has\(T\)\(String, Compare, T\)](#)

See Also

[VertexQueryContract Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Interval(T) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IQuery Interval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Public MustOverride Function Interval(Of T) (
    key As String,
    startValue As T,
    endValue As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Interval(
    String^ key,
    T startValue,
    T endValue
) abstract
```

F#

```
abstract Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

startValue

Type: **T**

[Missing <param name="startValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

endValue

Type: **T**

[Missing <param name="endValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Interval`1(System.String,`0,`0)"]

Implements

[IQuery.Interval\(T\)\(String, T, T\)](#)

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Labels Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Labels(System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Labels(  
    params string[] labels  
)
```

VB

```
Public Function Labels (  
    ParamArray labels As String()  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Labels(  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract Labels :  
    labels : string[] -> IVertexQuery  
override Labels :  
    labels : string[] -> IVertexQuery
```

Parameters

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Labels(System.String[])"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Labels(System.String[])"]

Implements

[IVertexQuery.Labels\(String\[\]\)](#)

VelocityDB Class Library

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Limit Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Limit(System.Int64)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IQuery Limit(  
    long max  
)
```

VB

```
Public MustOverride Function Limit (  
    max As Long  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Limit(  
    long long max  
) abstract
```

F#

```
abstract Limit :  
    max : int64 -> IQuery
```

Parameters

max

Type: [System.Int64](#)

[Missing <param name="max"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Limit(System.Int64)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Limit(System.Int64)"]

Implements

[IQuery.Limit\(Int64\)](#)

See Also

[VertexQueryContract Class](#)

VertexQueryContract.VertexIds Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.VertexIds"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Object> VertexIds ()
```

VB

```
Public Function VertexIds As IEnumerable(Of Object)
```

C++

```
public:  
virtual IEnumerable<Object^>^ VertexIds () sealed
```

F#

```
abstract VertexIds : unit -> IEnumerable<Object>  
override VertexIds : unit -> IEnumerable<Object>
```

Return Value

Type: [IEnumerable\(Object\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.VertexIds"]

Implements

[IVertexQuery.VertexIds\(\)](#)

See Also

[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)

VertexQueryContract.Vertices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Contracts](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IVertex> Vertices ()
```

VB

```
Public MustOverride Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^> Vertices () abstract
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Contracts.VertexQueryContract.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

See Also




[VertexQueryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Contracts Namespace](#)


VelocityGraph.Frontenac.Blueprints.Geo Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Geo"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | GeoCircle | |
|  | GeoPoint | |
|  | GeoRectangle | |

Interfaces

| | Interface | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | IGeoShape | |

GeoCircle Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class GeoCircle : IGeoShape
```

VB

```
<SerializableAttribute>
Public Class GeoCircle
    Implements IGeoShape
```

C++



```
[SerializableAttribute]
public ref class GeoCircle : IGeoShape
```

F#



```
[<SerializableAttribute>]
type GeoCircle =
    class
        interface IGeoShape
    end
```

The **GeoCircle** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------|----------------------------------------------------------|
|  | GeoCircle(GeoPoint, Double) | Initializes a new instance of the GeoCircle class |
|  | GeoCircle(Double, Double, Double) | Initializes a new instance of the GeoCircle class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-------------|
|  | Center | |
|  | Radius | |



VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------|
|  | GeoCircle(GeoPoint, Double) | Initializes a new instance of the GeoCircle class |
|  | GeoCircle(Double, Double, Double) | Initializes a new instance of the GeoCircle class |

See Also

[GeoCircle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle Constructor (GeoPoint, Double)

Initializes a new instance of the [GeoCircle](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoCircle(  
    GeoPoint center,  
    double radius  
)
```

VB

```
Public Sub New (  
    center As GeoPoint,  
    radius As Double  
)
```

C++

```
public:  
GeoCircle(  
    GeoPoint^ center,  
    double radius  
)
```

F#

```
new :  
    center : GeoPoint *  
    radius : float -> GeoCircle
```

Parameters

center

Type: [VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint](#)

[Missing <param name="center"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.#ctor(VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint,System.Double)"]

radius

Type: [System.Double](#)

[Missing <param name="radius"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.#ctor(VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint,System.Double)"]

See Also

[GeoCircle Class](#)

VelocityDB Class Library

[GeoCircle Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle Constructor (Double, Double, Double)

Initializes a new instance of the [GeoCircle](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoCircle(  
    double latitude,  
    double longitude,  
    double radius  
)
```

VB

```
Public Sub New (  
    latitude As Double,  
    longitude As Double,  
    radius As Double  
)
```

C++

```
public:  
GeoCircle(  
    double latitude,  
    double longitude,  
    double radius  
)
```

F#

```
new :  
    latitude : float *  
    longitude : float *  
    radius : float -> GeoCircle
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.#ctor(System.Double,System.Double,System.D
ouble)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.#ctor(System.Double,System.Double,System.D
ouble)"]

radius

Type: [System.Double](#)

[Missing <param name="radius"/> documentation for

**"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.#ctor(System.Double,System.Double,System.D
ouble)"]**

See Also

[GeoCircle Class](#)



[GeoCircle Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle.GeoCircle Properties

The [GeoCircle](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Center | |
|  | Radius | |

See Also

[GeoCircle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle.Center Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.Center"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoPoint Center { get; set; }
```

VB

```
Public Property Center As GeoPoint  
    Get  
    Set
```

C++

```
public:  
property GeoPoint^ Center {  
    GeoPoint^ get ();  
    void set (GeoPoint^ value);  
}
```

F#

```
member Center : GeoPoint with get, set
```

Property Value

Type: [GeoPoint](#)

See Also

[GeoCircle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoCircle.Radius Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Geo.GeoCircle.Radius"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public double Radius { get; set; }
```

VB

```
Public Property Radius As Double  
    Get  
    Set
```

C++

```
public:  
property double Radius {  
    double get ();  
    void set (double value);  
}
```

F#

```
member Radius : float with get, set
```

Property Value

Type: [Double](#)

See Also

[GeoCircle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class GeoPoint : IGeoShape
```

VB

```
<SerializableAttribute>
Public Class GeoPoint
    Implements IGeoShape
```

C++



```
[SerializableAttribute]
public ref class GeoPoint : IGeoShape
```

F#



```
[<SerializableAttribute>]
type GeoPoint =
    class
        interface IGeoShape
    end
```

The **GeoPoint** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|---------------------------------------------------------|
|  | GeoPoint() | Initializes a new instance of the GeoPoint class |
|  | GeoPoint(Double, Double) | Initializes a new instance of the GeoPoint class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | Latitude | |
|  | Longitude | |



VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|------------------------------------------------------------------|
|  | GeoPoint() | Initializes a new instance of the GeoPoint class |
|  | GeoPoint(Double, Double) | Initializes a new instance of the GeoPoint class |

See Also

[GeoPoint Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint Constructor

Initializes a new instance of the [GeoPoint](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoPoint ()
```

VB

```
Public Sub New
```

C++

```
public:  
GeoPoint ()
```

F#

```
new : unit -> GeoPoint
```

See Also

[GeoPoint Class](#)

[GeoPoint Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint Constructor (Double, Double)

Initializes a new instance of the [GeoPoint](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoPoint(  
    double latitude,  
    double longitude  
)
```

VB

```
Public Sub New (  
    latitude As Double,  
    longitude As Double  
)
```

C++

```
public:  
GeoPoint(  
    double latitude,  
    double longitude  
)
```

F#

```
new :  
    latitude : float *  
    longitude : float -> GeoPoint
```

Parameters

latitude

Type: [System.Double](#)

[Missing <param name="latitude"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint.#ctor(System.Double,System.Double)"]

longitude

Type: [System.Double](#)

[Missing <param name="longitude"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint.#ctor(System.Double,System.Double)"]

See Also

[GeoPoint Class](#)



[GeoPoint Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint.GeoPoint Properties

The [GeoPoint](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | Latitude | |
|  | Longitude | |

See Also

[GeoPoint Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint.Latitude Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint.Latitude"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public double Latitude { get; set; }
```

VB

```
Public Property Latitude As Double  
    Get  
    Set
```

C++

```
public:  
property double Latitude {  
    double get ();  
    void set (double value);  
}
```

F#

```
member Latitude : float with get, set
```

Property Value

Type: [Double](#)

See Also

[GeoPoint Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoPoint.Longitude Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint.Longitude"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public double Longitude { get; set; }
```

VB

```
Public Property Longitude As Double  
    Get  
    Set
```

C++

```
public:  
property double Longitude {  
    double get ();  
    void set (double value);  
}
```

F#

```
member Longitude : float with get, set
```

Property Value

Type: [Double](#)

See Also

[GeoPoint Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class GeoRectangle : IGeoShape
```

VB

```
<SerializableAttribute>
Public Class GeoRectangle
    Implements IGeoShape
```

C++



```
[SerializableAttribute]
public ref class GeoRectangle : IGeoShape
```

F#

```
[<SerializableAttribute>]
type GeoRectangle =
    class
        interface IGeoShape
    end
```

The **GeoRectangle** type exposes the following members.

Constructors

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
|  GeoRectangle(GeoPoint, GeoPoint) | Initializes a new instance of the GeoRectangle class |
|  GeoRectangle(Double, Double, Double, Double) | Initializes a new instance of the GeoRectangle class |

Properties

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------|
|  BottomRight | |





[TopLeft](#)

See Also

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------------------|
|  | GeoRectangle(GeoPoint, GeoPoint) | Initializes a new instance of the GeoRectangle class |
|  | GeoRectangle(Double, Double, Double, Double) | Initializes a new instance of the GeoRectangle class |

See Also

[GeoRectangle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle Constructor (GeoPoint, GeoPoint)

Initializes a new instance of the [GeoRectangle](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoRectangle(  
    GeoPoint topLeft,  
    GeoPoint bottomRight  
)
```

VB

```
Public Sub New (  
    topLeft As GeoPoint,  
    bottomRight As GeoPoint  
)
```

C++

```
public:  
GeoRectangle(  
    GeoPoint^ topLeft,  
    GeoPoint^ bottomRight  
)
```

F#

```
new :  
    topLeft : GeoPoint *  
    bottomRight : GeoPoint -> GeoRectangle
```

Parameters

topLeft

Type: [VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint](#)

[Missing <param name="topLeft"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint,VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint)"]

bottomRight

Type: [VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint](#)

[Missing <param name="bottomRight"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint,VelocityGraph.Frontenac.Blueprints.Geo.GeoPoint)"]

See Also

[GeoRectangle Class](#)

VelocityDB Class Library

[GeoRectangle Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle Constructor (Double, Double, Double, Double)

Initializes a new instance of the [GeoRectangle](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoRectangle(  
    double minX,  
    double maxX,  
    double minY,  
    double maxY  
)
```

VB

```
Public Sub New (  
    minX As Double,  
    maxX As Double,  
    minY As Double,  
    maxY As Double  
)
```

C++

```
public:  
GeoRectangle(  
    double minX,  
    double maxX,  
    double minY,  
    double maxY  
)
```

F#

```
new :  
    minX : float *  
    maxX : float *  
    minY : float *  
    maxY : float -> GeoRectangle
```

Parameters

minX

Type: [System.Double](#)

[Missing <param name="minX"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(System.Double,System.Double,System.Double,System.Double)"]

maxX

Type: [System.Double](#)

[Missing <param name="maxX"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(System.Double,System.Double,System.Double,System.Double)"]

minY

Type: [System.Double](#)

[Missing <param name="minY"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(System.Double,System.Double,System.Double,System.Double)"]

maxY

Type: [System.Double](#)

[Missing <param name="maxY"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.#ctor(System.Double,System.Double,System.Double,System.Double)"]

See Also

[GeoRectangle Class](#)

[GeoRectangle Overload](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle.GeoRectangle Properties

The [GeoRectangle](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | BottomRight | |
|  | TopLeft | |

See Also

[GeoRectangle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle.BottomRight Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.BottomRight"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoPoint BottomRight { get; set; }
```

VB

```
Public Property BottomRight As GeoPoint  
    Get  
    Set
```

C++

```
public:  
property GeoPoint^ BottomRight {  
    GeoPoint^ get ();  
    void set (GeoPoint^ value);  
}
```

F#

```
member BottomRight : GeoPoint with get, set
```

Property Value

Type: [GeoPoint](#)

See Also

[GeoRectangle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

GeoRectangle.TopLeft Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Geo.GeoRectangle.TopLeft"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GeoPoint TopLeft { get; set; }
```

VB

```
Public Property TopLeft As GeoPoint  
    Get  
    Set
```

C++

```
public:  
property GeoPoint^ TopLeft {  
    GeoPoint^ get ();  
    void set (GeoPoint^ value);  
}
```

F#

```
member TopLeft : GeoPoint with get, set
```

Property Value

Type: [GeoPoint](#)

See Also

[GeoRectangle Class](#)

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

IGeoShape Interface

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Geo.IGeoShape"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Geo](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IGeoShape
```

VB

```
Public Interface IGeoShape
```

C++

```
public interface class IGeoShape
```

F#

```
type IGeoShape = interface end
```





See Also

[VelocityGraph.Frontenac.Blueprints.Geo Namespace](#)

VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Impls.TG"]

Classes

| Class | Description |
|-------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
|  TinkerGraph | An in-memory, reference implementation of the property TinkerGraph interfaces provided by Blueprints. |
|  TinkerGraphFactory | |
|  TinkerMetadataReader | Reads TinkerGraph metadata from a Stream. |
|  TinkerStorageContract | |

Enumerations

| Enumeration | Description |
|------------------------------------------------------------------------------------------------------------------------|-------------|
|  TinkerGraph.FileType | |

TinkerGraph Class

An in-memory, reference implementation of the property TinkerGraph interfaces provided by Blueprints.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
[SerializableAttribute]
public class TinkerGraph : IIndexableGraph,
    IGraph, IKeyIndexableGraph
```

VB

```
<SerializableAttribute>
Public Class TinkerGraph
    Implements IIndexableGraph, IGraph, IKeyIndexableGraph
```

C++




```
[SerializableAttribute]
public ref class TinkerGraph : IIndexableGraph,
    IGraph, IKeyIndexableGraph
```

F#


```
[<SerializableAttribute>]
type TinkerGraph =
    class
        interface IIndexableGraph
        interface IGraph
        interface IKeyIndexableGraph
    end
```

The **TinkerGraph** type exposes the following members.

Constructors

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|
|  TinkerGraph() | Initializes a new instance of the TinkerGraph class |
|  TinkerGraph(String) | Initializes a new instance of the TinkerGraph class |
|  TinkerGraph(String, TinkerGraph.FileType) | Initializes a new instance of the TinkerGraph class |



Properties













| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Clear | |
|  | CreateIndex | |
|  | CreateKeyIndex | |
|  | DropIndex | |
|  | DropKeyIndex | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetIndex | |
|  | GetIndexedKeys | |
|  | GetIndices | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |




| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------------|
|  | TinkerGraph() | Initializes a new instance of the TinkerGraph class |
|  | TinkerGraph(String) | Initializes a new instance of the TinkerGraph class |
|  | TinkerGraph(String, TinkerGraph.FileType) | Initializes a new instance of the TinkerGraph class |

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph Constructor

Initializes a new instance of the [TinkerGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TinkerGraph ()
```

VB

```
Public Sub New
```

C++

```
public:  
TinkerGraph ()
```

F#

```
new : unit -> TinkerGraph
```

See Also

[TinkerGraph Class](#)

[TinkerGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph Constructor (String)

Initializes a new instance of the [TinkerGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TinkerGraph(  
    string directory  
)
```

VB

```
Public Sub New (  
    directory As String  
)
```

C++

```
public:  
TinkerGraph(  
    String^ directory  
)
```

F#

```
new :  
    directory : string -> TinkerGraph
```

Parameters

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.#ctor(System.String)"]

See Also

[TinkerGraph Class](#)

[TinkerGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph Constructor (String, TinkerGraph.FileType)

Initializes a new instance of the [TinkerGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TinkerGraph(  
    string directory,  
    TinkerGraph.FileType fileType  
)
```

VB

```
Public Sub New (  
    directory As String,  
    fileType As TinkerGraph.FileType  
)
```

C++

```
public:  
TinkerGraph(  
    String^ directory,  
    TinkerGraph.FileType fileType  
)
```

F#

```
new :  
    directory : string *  
    fileType : TinkerGraph.FileType -> TinkerGraph
```

Parameters

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.#ctor(System.String,VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.FileType)"]

fileType

Type: [VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.FileType](#)

[Missing <param name="fileType"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.#ctor(System.String,VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.FileType)"]

See Also

[TinkerGraph Class](#)

VelocityDB Class Library


[TinkerGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.TinkerGraph Properties

The [TinkerGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.Features Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual Features Features { get; }
```

VB

```
Public Overridable ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get ();  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)




TinkerGraph.TinkerGraph Methods












The [TinkerGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Clear | |
|  | CreateIndex | |
|  | CreateKeyIndex | |
|  | DropIndex | |
|  | DropKeyIndex | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetIndex | |
|  | GetIndexedKeys | |
|  | GetIndices | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEdge AddEdge (  
    Object id,  
    IVertex outVertex,  
    IVertex inVertex,  
    string label  
)
```

VB

```
Public Overridable Function AddEdge (  
    id As Object,  
    outVertex As IVertex,  
    inVertex As IVertex,  
    label As String  
) As IEdge
```

C++

```
public:  
virtual IEdge^ AddEdge (  
    Object^ id,  
    IVertex^ outVertex,  
    IVertex^ inVertex,  
    String^ label  
)
```

F#

```
abstract AddEdge :  
    id : Object *  
    outVertex : IVertex *  
    inVertex : IVertex *  
    label : string -> IEdge  
override AddEdge :  
    id : Object *  
    outVertex : IVertex *  
    inVertex : IVertex *  
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Overridable Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
)
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.AddVertex(System.Object)"]

Implements

[IGraph.AddVertex\(Object\)](#)

VelocityDB Class Library

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.Clear Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.Clear"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Clear()
```

VB

```
Public Sub Clear
```

C++

```
public:  
void Clear()
```

F#

```
member Clear : unit -> unit
```

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.CreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Overridable Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.CreateKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void CreateKeyIndex(  
    string key,  
    Type elementClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Overridable Sub CreateKeyIndex (  
    key As String,  
    elementClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
public:  
virtual void CreateKeyIndex(  
    String^ key,  
    Type^ elementClass,  
    ... array<Parameter^>^ indexParameters  
)
```

F#

```
abstract CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit  
override CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IKeyIndexableGraph.CreateKeyIndex\(String, Type,Parameter\[\]\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.DropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.DropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void DropIndex(  
    string indexName  
)
```

VB

```
Public Overridable Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
)
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.DropIndex(System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.DropKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.DropKeyIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void DropKeyIndex(  
    string key,  
    Type elementClass  
)
```

VB

```
Public Overridable Sub DropKeyIndex (  
    key As String,  
    elementClass As Type  
)
```

C++

```
public:  
virtual void DropKeyIndex(  
    String^ key,  
    Type^ elementClass  
)
```

F#

```
abstract DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit  
override DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.DropKeyIndex(System.String,System.Type)"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.DropKeyIndex(System.String,System.Type)"]

Implements

[IKeyIndexableGraph.DropKeyIndex\(String, Type\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEdge GetEdge(  
    Object id  
)
```

VB

```
Public Overridable Function GetEdge (  
    id As Object  
) As IEdge
```

C++

```
public:  
virtual IEdge^ GetEdge(  
    Object^ id  
)
```

F#

```
abstract GetEdge :  
    id : Object -> IEdge  
override GetEdge :  
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetEdge(System.Object)"]

Implements

[IGraph.GetEdge\(Object\)](#)

VelocityDB Class Library

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[TinkerGraph^h Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetEdges Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Overridable Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdgeh> GetEdges ()
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[TinkerGraph^h Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IEdge> GetEdges (
    string key,
    Object value
)
```

VB

```
Public Overridable Function GetEdges (
    key As String,
    value As Object
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    String^ key,
    Object^ value
)
```

F#

```
abstract GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
override GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[TinkerGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetIndex(System.String,System.Type)"
]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Overridable Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
)
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetIndex(System.String,System.Type)"
]

indexClass

Type: [System.Type](#)

**[Missing <param name="indexClass"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetIndex(System.String,System.Type)"
]**

Return Value

Type: [IIndex](#)

**[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetIndex(System.String,System.Type)"
]**

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph[^].GetIndexedKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph[^].GetIndexedKeys(System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<string> GetIndexedKeys (
    Type elementClass
)
```

VB

```
Public Overridable Function GetIndexedKeys (
    elementClass As Type
) As IEnumerable(Of String)
```

C++

```
public:
virtual IEnumerable<String^>^ GetIndexedKeys (
    Type^ elementClass
)
```

F#

```
abstract GetIndexedKeys :
    elementClass : Type -> IEnumerable<string>
override GetIndexedKeys :
    elementClass : Type -> IEnumerable<string>
```

Parameters

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph[^].GetIndexedKeys(System.Type)"]

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph[^].GetIndexedKeys(System.Type)"]

Implements

[IKeyIndexableGraph.GetIndexedKeys\(Type\)](#)

VelocityDB Class Library

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetIndices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetIndices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Overridable Function GetIndices As IEnumerable(Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndexh> GetIndices ()
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetIndices"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[TinkerGraph^h Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Overridable Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
)
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetVertex(System.Object)"]

Implements

[IGraph.GetVertex\(Object\)](#)

VelocityDB Class Library

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph[^].GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[TinkerGraph[^] Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.GetVertices Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Overridable Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertexh>h GetVertices ()
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[TinkerGraph^h Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public Overridable Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
)
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
override GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[TinkerGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IQuery Query()
```

VB

```
Public Overridable Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query()
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Public Overridable Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void RemoveEdge (  
    IEdge^ edge  
)
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit  
override RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Overridable Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
)
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.Shutdown Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph.ToString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph.ToString"]

See Also

[TinkerGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraph^h.FileType Enumeration

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph^h.FileType"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public enum FileType
```

```
VB  
Public Enumeration FileType
```

```
C++  
public enum class FileType
```

```
F#  
type FileType
```

Members

| Member name | Value | Description |
|-----------------|-------|-------------|
| DotNet | 0 | |
| Gml | 1 | |
| Graphml | 2 | |
| Graphson | 3 | |

See Also

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraphFactory Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class TinkerGraphFactory
```

VB

```
Public NotInheritable Class TinkerGraphFactory
```

C++

```
public ref class TinkerGraphFactory abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type TinkerGraphFactory = class end
```

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|-------------|
|  | CreateTinkerGraph() | |
|  | CreateTinkerGraph(IGraph) | |

See Also

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraphFactory.TinkerGraphFactory Methods

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|-------------|
|  | CreateTinkerGraph() | |
|  | CreateTinkerGraph(IGraph) | |

See Also

[TinkerGraphFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraphFactory.CreateTinkerGraph Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|-------------|
|  | CreateTinkerGraph() | |
|  | CreateTinkerGraph(IGraph) | |

See Also

[TinkerGraphFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraphFactory.CreateTinkerGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory.CreateTinkerGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static TinkerGraph CreateTinkerGraph()
```

VB

```
Public Shared Function CreateTinkerGraph As TinkerGraph
```

C++

```
public:  
static TinkerGraph^ CreateTinkerGraph()
```

F#

```
static member CreateTinkerGraph : unit -> TinkerGraph
```

Return Value

Type: [TinkerGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory.CreateTinkerGraph"]

See Also

[TinkerGraphFactory Class](#)

[CreateTinkerGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerGraphFactory.CreateTinkerGraph Method (IGraph)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory.CreateTinkerGraph(VelocityGraph.Frontenac.Blueprints.IGraph)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void CreateTinkerGraph (
    IGraph graph
)
```

VB

```
Public Shared Sub CreateTinkerGraph (
    graph As IGraph
)
```

C++

```
public:
static void CreateTinkerGraph (
    IGraph^ graph
)
```

F#

```
static member CreateTinkerGraph :
    graph : IGraph -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraphFactory.CreateTinkerGraph(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also

[TinkerGraphFactory Class](#)

[CreateTinkerGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader Class

Reads TinkerGraph metadata from a Stream.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerMetadataReader

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class TinkerMetadataReader
```

VB

```
Public Class TinkerMetadataReader
```

C++


```
public ref class TinkerMetadataReader
```

F#





```
type TinkerMetadataReader = class end
```

The **TinkerMetadataReader** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|---------------------------------------------------------------------|
|  | TinkerMetadataReader | Initializes a new instance of the TinkerMetadataReader class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------|
|  | Load(Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(String) | Read TinkerGraph metadata from a file. |
|  | Load(TinkerGraph, Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(TinkerGraph, String) | Read TinkerGraph metadata from a file. |

See Also

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader Constructor

Initializes a new instance of the [TinkerMetadataReader](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TinkerMetadataReader(  
    TinkerGraph tinkerGraph  
)
```

VB

```
Public Sub New (  
    tinkerGraph As TinkerGraph  
)
```

C++

```
public:  
TinkerMetadataReader(  
    TinkerGraph^ tinkerGraph  
)
```

F#

```
new :  
    tinkerGraph : TinkerGraph -> TinkerMetadataReader
```

Parameters

tinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph](#)

[Missing <param name="tinkerGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerMetadataReader.#ctor(VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph)"]





See Also

[TinkerMetadataReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.TinkerMetadataReader Methods

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------|
|  | Load(Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(String) | Read TinkerGraph metadata from a file. |
|  | Load(TinkerGraph, Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(TinkerGraph, String) | Read TinkerGraph metadata from a file. |





See Also

[TinkerMetadataReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.Load Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------|
|  | Load(Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(String) | Read TinkerGraph metadata from a file. |
|  | Load(TinkerGraph, Stream) | Read TinkerGraph metadata from a Stream. |
|  | Load(TinkerGraph, String) | Read TinkerGraph metadata from a file. |

See Also

[TinkerMetadataReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.Load Method (Stream)

Read TinkerGraph metadata from a Stream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Load(  
    Stream inputStream  
)
```

VB

```
Public Sub Load (  
    inputStream As Stream  
)
```

C++

```
public:  
void Load(  
    Stream^ inputStream  
)
```

F#

```
member Load :  
    inputStream : Stream -> unit
```

Parameters

inputStream

Type: [System.IO.Stream](#)

the Stream to read the TinkerGraph metadata from

See Also

[TinkerMetadataReader Class](#)

[Load Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.Load Method (String)

Read TinkerGraph metadata from a file.

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Load(  
    string filename  
)
```

VB

```
Public Sub Load (  
    filename As String  
)
```

C++

```
public:  
void Load(  
    String^ filename  
)
```

F#

```
member Load :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

the name of the file to read the TinkerGraph metadata from

See Also

[TinkerMetadataReader Class](#)

[Load Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.Load Method (TinkerGraph, Stream)

Read TinkerGraph metadata from a Stream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void Load(  
    TinkerGraph tinkerGraph,  
    Stream inputStream  
)
```

VB

```
Public Shared Sub Load (  
    tinkerGraph As TinkerGraph,  
    inputStream As Stream  
)
```

C++

```
public:  
static void Load(  
    TinkerGraph^ tinkerGraph,  
    Stream^ inputStream  
)
```

F#

```
static member Load :  
    tinkerGraph : TinkerGraph *  
    inputStream : Stream -> unit
```

Parameters

tinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph](#)

the IGraph to push the metadata to

inputStream

Type: [System.IO.Stream](#)

the Stream to read the TinkerGraph metadata from

See Also

[TinkerMetadataReader Class](#)

[Load Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerMetadataReader.Load Method (TinkerGraph, String)

Read TinkerGraph metadata from a file.

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void Load(  
    TinkerGraph tinkerGraph,  
    string filename  
)
```

VB

```
Public Shared Sub Load (  
    tinkerGraph As TinkerGraph,  
    filename As String  
)
```

C++

```
public:  
static void Load(  
    TinkerGraph^ tinkerGraph,  
    String^ filename  
)
```

F#

```
static member Load :  
    tinkerGraph : TinkerGraph *  
    filename : string -> unit
```

Parameters

tinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph](#)

the TinkerGraph to push the data to

filename

Type: [System.String](#)

the name of the file to read the TinkerGraph metadata from

See Also

[TinkerMetadataReader Class](#)

[Load Overload](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerStorageContract Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class TinkerStorageContract
```

VB

```
Public MustInherit Class TinkerStorageContract
```

C++



```
public ref class TinkerStorageContract abstract
```

F#

```
[<AbstractClassAttribute>]  
type TinkerStorageContract = class end
```

The **TinkerStorageContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Load | |
|  | Save | |



See Also

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerStorageContract.TinkerStorageContract Methods

The [TinkerStorageContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Load | |
|  | Save | |

See Also

[TinkerStorageContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerStorageContract.Load Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Load(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public TinkerGraph Load(  
    string directory  
)
```

VB

```
Public Function Load (  
    directory As String  
) As TinkerGraph
```

C++

```
public:  
virtual TinkerGraph^ Load(  
    String^ directory  
) sealed
```

F#

```
abstract Load :  
    directory : string -> TinkerGraph  
override Load :  
    directory : string -> TinkerGraph
```

Parameters

directory

Type: [System.String](#)

[Missing <param name="directory"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Load(System.String)"]

Return Value

Type: [TinkerGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Load(System.String)"]

See Also

[TinkerStorageContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

TinkerStorageContract.Save Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Save(VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph, System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Impls.TG](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Save(  
    TinkerGraph tinkerGraph,  
    string directory  
)
```

VB

```
Public Sub Save (  
    tinkerGraph As TinkerGraph,  
    directory As String  
)
```

C++

```
public:  
virtual void Save(  
    TinkerGraph^ tinkerGraph,  
    String^ directory  
) sealed
```

F#

```
abstract Save :  
    tinkerGraph : TinkerGraph *  
    directory : string -> unit  
override Save :  
    tinkerGraph : TinkerGraph *  
    directory : string -> unit
```

Parameters

tinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph](#)

[Missing <param name="tinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Save(VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerGraph, System.String)"]

directory

Type: [System.String](#)

**[Missing <param name="directory"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Impls.TG.TinkerStorageContract.Save(VelocityGraph.Frontena
c.Blueprints.Impls.TG.TinkerGraph, System.String)"]**

See Also

















[TinkerStorageContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Impls.TG Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util"]

Classes

| Class | Description |
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  DefaultGraphQuery | For those graph engines that do not support the low-level querying of the vertices or edges, then DefaultQuery can be used. DefaultQuery assumes, at minimum, that Graph.getVertices() and Graph.getEdges() is implemented by the respective Graph. |
|  DefaultQuery | |
|  DefaultQuery.HasContainer | |
|  DefaultVertexQuery | For those graph engines that do not support the low-level querying of the edges of a vertex, then DefaultVertexQuery can be used. DefaultVertexQuery assumes, at minimum, that Vertex.getOutEdges() and Vertex.getInEdges() is implemented by the respective Vertex. |
|  EdgeHelpers | |
|  ElementHelpers | |
|  ExceptionFactory | The ExceptionFactory provides standard exceptions that should be used by all Blueprints implementations. This ensures that the look-and-feel of all implementations are the same in terms of terminology and punctuation. |
|  GraphHelpers | |
|  IndexableGraphHelpers | |
|  KeyIndexableGraphHelpers | |
|  Multiterable(TS) | A helper class that is used to combine multiple iterables into a single closeable IEnumerable. |
|  PropertyFilteredIterable(T) | This is a helper class for filtering an IEnumerable of elements by their key/value. Useful for Graph implementations that do not support automatic key indices and need to filter on Graph.getVertices/Edges(key,value). |
|  StringFactory | A collection of helpful methods for creating standard toString() representations of graph-related objects. |
|  VertexHelpers | |
|  VerticesFromEdgesIterable | VerticesFromEdgesIterable is a helper class that returns vertices that meet the direction/label criteria of the incident edges. |
|  WrappingCloseableIterable(T) | |

DefaultGraphQuery Class

For those graph engines that do not support the low-level querying of the vertices or edges, then DefaultQuery can be used. DefaultQuery assumes, at minimum, that Graph.getVertices() and Graph.getEdges() is implemented by the respective Graph.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery](#)

VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class DefaultGraphQuery : DefaultQuery
```

VB

```
Public Class DefaultGraphQuery
    Inherits DefaultQuery
```

C++


```
public ref class DefaultGraphQuery : public DefaultQuery
```

F#



```
type DefaultGraphQuery =
    class
        inherit DefaultQuery
    end
```

The **DefaultGraphQuery** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | DefaultGraphQuery | Initializes a new instance of the DefaultGraphQuery class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|--------------------------------------------------------|
|  | Edges | (Overrides DefaultQuery.Edges(.) .) |
|  | Vertices | (Overrides DefaultQuery.Vertices(.) .) |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultGraphQuery Constructor

Initializes a new instance of the [DefaultGraphQuery](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DefaultGraphQuery(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
DefaultGraphQuery(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> DefaultGraphQuery
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also



[DefaultGraphQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultGraphQuery.DefaultGraphQuery Methods

The [DefaultGraphQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------|
|  | Edges | (Overrides DefaultQuery.Edges() .) |
|  | Vertices | (Overrides DefaultQuery.Vertices() .) |

See Also

[DefaultGraphQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultGraphQuery.Edges Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<IEdge> Edges ()
```

VB

```
Public Overrides Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^> Edges () override
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>  
override Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery.Edges"]

Implements

[IQuery.Edges\(\)](#)

See Also

[DefaultGraphQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultGraphQuery.Vertices Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<IVertex> Vertices ()
```

VB

```
Public Overrides Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ Vertices () override
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>  
override Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

See Also

[DefaultGraphQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery

[VelocityGraph.Frontenac.Blueprints.Util.DefaultGraphQuery](#)

[VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class DefaultQuery : IQuery
```

VB

```
Public MustInherit Class DefaultQuery
    Implements IQuery
```

C++







```
public ref class DefaultQuery abstract : IQuery
```

F#



```
[<AbstractClassAttribute>]
type DefaultQuery =
    class
        interface IQuery
    end
```

The **DefaultQuery** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Limit | |
|  | Vertices | |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | Direction | |
|  | Labels | |







See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.DefaultQuery Methods

The [DefaultQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Limit | |
|  | Vertices | |

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Edges Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IEdge> Edges ()
```

VB

```
Public MustOverride Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ Edges () abstract
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Edges"]

Implements

[IQuery.Edges\(\)](#)

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Has Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Has Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IQuery Has(  
    string key,  
    Object value  
)
```

VB

```
Public Overridable Function Has (  
    key As String,  
    value As Object  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Has(  
    String^ key,  
    Object^ value  
)
```

F#

```
abstract Has :  
    key : string *  
    value : Object -> IQuery  
override Has :  
    key : string *  
    value : Object -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has(System.String,System.Object)"]

VelocityDB Class Library

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has(System.String,System.Object)"]

Implements

[IQuery.Has\(String, Object\)](#)

See Also

[DefaultQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Has(*T*) Method (String, Compare, *T*)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IQuery Has<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Public Overridable Function Has(Of T) (
    key As String,
    compare As Compare,
    value As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Has(
    String^ key,
    Compare compare,
    T value
)
```

F#

```
abstract Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
override Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

value

Type: *T*

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Implements

[IQuery.Has\(T\)\(String, Compare, T\)](#)

See Also

[DefaultQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Interval(T) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IQuery Interval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Public Overridable Function Interval(Of T) (
    key As String,
    startValue As T,
    endValue As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Interval(
    String^ key,
    T startValue,
    T endValue
)
```

F#

```
abstract Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
override Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]

startValue

Type: **T**

**[Missing <param name="startValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]**

endValue

Type: **T**

**[Missing <param name="endValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]**

Type Parameters

T

**[Missing <typeparam name="T"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]**

Return Value

Type: [IQuery](#)

**[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Interval`1(System.String,`0,`0)"]**

Implements

[IQuery.Interval\(T\)\(String, T, T\)](#)

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Limit Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Limit(System.Int64)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual IQuery Limit(  
    long max  
)
```

VB

```
Public Overridable Function Limit (  
    max As Long  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Limit(  
    long long max  
)
```

F#

```
abstract Limit :  
    max : int64 -> IQuery  
override Limit :  
    max : int64 -> IQuery
```

Parameters

max

Type: [System.Int64](#)

[Missing <param name="max"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Limit(System.Int64)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Limit(System.Int64)"]

Implements

[IQuery.Limit\(Int64\)](#)

VelocityDB Class Library

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Vertices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract IEnumerable<IVertex> Vertices ()
```

VB

```
Public MustOverride Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ Vertices () abstract
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

See Also



[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.DefaultQuery Fields

The [DefaultQuery](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | Direction | |
|  | Labels | |

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Direction Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Direction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Direction Direction
```

VB

```
Public Direction As Direction
```

C++

```
public:  
Direction Direction
```

F#

```
val mutable Direction: Direction
```

Field Value

Type: [Direction](#)

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.Labels Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.Labels"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string[] Labels
```

VB

```
Public Labels As String()
```

C++

```
public:  
array<String^>^ Labels
```

F#

```
val mutable Labels: string[]
```

Field Value

Type: [String\[\]](#)

See Also

[DefaultQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class HasContainer
```

VB

```
Public Class HasContainer
```

C++


```
public ref class HasContainer
```

F#


```
type HasContainer = class end
```

The DefaultQuery.HasContainer type exposes the following members.




Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------------------|
|  | DefaultQuery.HasContainer | Initializes a new instance of the DefaultQuery.HasContainer class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-------------|
|  | IsLegal | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |
|  | Key | |
|  | Value | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer Constructor

Initializes a new instance of the [DefaultQuery.HasContainer](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public HasContainer(  
    string key,  
    Object value,  
    Compare compare  
)
```

VB

```
Public Sub New (  
    key As String,  
    value As Object,  
    compare As Compare  
)
```

C++

```
public:  
HasContainer(  
    String^ key,  
    Object^ value,  
    Compare compare  
)
```

F#

```
new :  
    key : string *  
    value : Object *  
    compare : Compare -> HasContainer
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.#ctor(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.Compare)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.#ctor(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.Compare)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.#ctor(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.Compare)"]

See Also


[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

HasContainer.HasContainer Methods

The [DefaultQuery.HasContainer](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------|
|  | IsLegal | |

See Also

[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer.IsLegal Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.IsLegal(VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsLegal(  
    IElement element  
)
```

VB

```
Public Function IsLegal (  
    element As IElement  
) As Boolean
```

C++

```
public:  
bool IsLegal(  
    IElement^ element  
)
```

F#

```
member IsLegal :  
    element : IElement -> bool
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.IsLegal(VelocityGraph.Frontenac.Blueprints.IElement)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.IsLegal(VelocityGraph.Frontenac.Blueprints.IElement)"]

See Also




[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

HasContainer.HasContainer Fields

The [DefaultQuery.HasContainer](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |
|  | Key | |
|  | Value | |

See Also

[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer.Compare Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.Compare"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Compare Compare
```

VB

```
Public Compare As Compare
```

C++

```
public:  
Compare Compare
```

F#

```
val mutable Compare: Compare
```

Field Value

Type: [Compare](#)

See Also

[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer.Key Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.Key"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Key
```

VB

```
Public Key As String
```

C++

```
public:  
String^ Key
```

F#

```
val mutable Key: string
```

Field Value

Type: [String](#)

See Also

[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultQuery.HasContainer.Value Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery.HasContainer.Value"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object Value
```

VB

```
Public Value As Object
```

C++

```
public:  
Object^ Value
```

F#

```
val mutable Value: Object
```

Field Value

Type: [Object](#)

See Also

[DefaultQuery.HasContainer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery Class

For those graph engines that do not support the low-level querying of the edges of a vertex, then `DefaultVertexQuery` can be used. `DefaultVertexQuery` assumes, at minimum, that `Vertex.getOutEdges()` and `Vertex.getInEdges()` is implemented by the respective `Vertex`.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.DefaultQuery](#)

`VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery`

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class DefaultVertexQuery : DefaultQuery,
    IVertexQuery, IQuery
```

VB

```
Public Class DefaultVertexQuery
    Inherits DefaultQuery
    Implements IVertexQuery, IQuery
```

C++


```
public ref class DefaultVertexQuery : public DefaultQuery,
    IVertexQuery, IQuery
```

F#







```
type DefaultVertexQuery =
    class
        inherit DefaultQuery
        interface IVertexQuery
        interface IQuery
    end
```

The **DefaultVertexQuery** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------|
|  | DefaultVertexQuery | Initializes a new instance of the DefaultVertexQuery class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------|
|  | Count | |
|  | Direction | |
|  | Edges | (Overrides DefaultQuery.Edges() .) |
|  | Labels | |
|  | VertexIds | |
|  | Vertices | (Overrides DefaultQuery.Vertices() .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery Constructor

Initializes a new instance of the [DefaultVertexQuery](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public DefaultVertexQuery(  
    IVertex vertex  
)
```

VB

```
Public Sub New (  
    vertex As IVertex  
)
```

C++

```
public:  
DefaultVertexQuery(  
    IVertex^ vertex  
)
```

F#

```
new :  
    vertex : IVertex -> DefaultVertexQuery
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also







[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.DefaultVertexQuery Methods

The [DefaultVertexQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------|
|  | Count | |
|  | Direction | |
|  | Edges | (Overrides DefaultQuery.Edges() .) |
|  | Labels | |
|  | VertexIds | |
|  | Vertices | (Overrides DefaultQuery.Vertices() .) |

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.Count Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Count"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count ()
```

VB

```
Public Function Count As Long
```

C++

```
public:  
virtual long long Count () sealed
```

F#

```
abstract Count : unit -> int64  
override Count : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Count"]

Implements

[IVertexQuery.Count\(\)](#)

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.Direction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Direction(  
    Direction direction  
)
```

VB

```
Public Function Direction (  
    direction As Direction  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Direction(  
    Direction direction  
) sealed
```

F#

```
abstract Direction :  
    direction : Direction -> IVertexQuery  
override Direction :  
    direction : Direction -> IVertexQuery
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IVertexQuery.Direction\(Direction\)](#)

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.Edges Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<IEdge> Edges ()
```

VB

```
Public Overrides Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^> Edges () override
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>  
override Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Edges"]

Implements

[IQuery.Edges\(\)](#)

[IQuery.Edges\(\)](#)

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.Labels Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Labels(System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Labels(  
    params string[] labels  
)
```

VB

```
Public Function Labels (  
    ParamArray labels As String()  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Labels(  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract Labels :  
    labels : string[] -> IVertexQuery  
override Labels :  
    labels : string[] -> IVertexQuery
```

Parameters

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Labels(System.String[])"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Labels(System.String[])"]

Implements

[IVertexQuery.Labels\(String\[\]\)](#)

VelocityDB Class Library

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.VertexIds Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.VertexIds"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Object> VertexIds ()
```

VB

```
Public Function VertexIds As IEnumerable(Of Object)
```

C++

```
public:  
virtual IEnumerable<Object^>^ VertexIds () sealed
```

F#

```
abstract VertexIds : unit -> IEnumerable<Object>  
override VertexIds : unit -> IEnumerable<Object>
```

Return Value

Type: [IEnumerable\(Object\)](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.VertexIds"]

Implements

[IVertexQuery.VertexIds\(\)](#)

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

DefaultVertexQuery.Vertices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<IVertex> Vertices ()
```

VB

```
Public Overrides Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ Vertices () override
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>  
override Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.DefaultVertexQuery.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

[IQuery.Vertices\(\)](#)

See Also

[DefaultVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

EdgeHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.EdgeHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.EdgeHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class EdgeHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class EdgeHelpers
```

C++





```
[ExtensionAttribute]  
public ref class EdgeHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type EdgeHelpers = class end
```

The **EdgeHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|   | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. |
|   | RelabelEdges | Edges are relabeled by creating new edges with the same properties, but new label. Note that for each edge is deleted and an edge is added. |





See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

EdgeHelpers.EdgeHelpers Methods

The [EdgeHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|   | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. |
|   | RelabelEdges | Edges are relabeled by creating new edges with the same properties, but new label. Note that for each edge is deleted and an edge is added. |

See Also

[EdgeHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

EdgeHelpers.RelabelEdge Method

An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEdge RelabelEdge(
    this IEdge oldEdge,
    IGraph graph,
    Object newId,
    string newLabel
)
```

VB

```
<ExtensionAttribute>
Public Shared Function RelabelEdge (
    oldEdge As IEdge,
    graph As IGraph,
    newId As Object,
    newLabel As String
) As IEdge
```

C++

```
public:
[ExtensionAttribute]
static IEdge^ RelabelEdge(
    IEdge^ oldEdge,
    IGraph^ graph,
    Object^ newId,
    String^ newLabel
)
```

F#

```
[<ExtensionAttribute>]
static member RelabelEdge :
    oldEdge : IEdge *
    graph : IGraph *
    newId : Object *
    newLabel : string -> IEdge
```

Parameters

oldEdge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the existing edge to "relabel"

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)
the graph to add the new edge to

newId

Type: [System.Object](#)
the id of the new edge

newLabel

Type: [System.String](#)
the label of the new edge

Return Value

Type: [IEdge](#)
the newly created edge

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEdge](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[EdgeHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

EdgeHelpers.RelabelEdges Method

Edges are relabeled by creating new edges with the same properties, but new label. Note that for each edge is deleted and an edge is added.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void RelabelEdges (
    this IEnumerable<IEdge> oldEdges,
    IGraph graph,
    string newLabel
)
```

VB

```
<ExtensionAttribute>
Public Shared Sub RelabelEdges (
    oldEdges As IEnumerable(Of IEdge),
    graph As IGraph,
    newLabel As String
)
```

C++

```
public:
[ExtensionAttribute]
static void RelabelEdges (
    IEnumerable<IEdge^>^ oldEdges,
    IGraph^ graph,
    String^ newLabel
)
```

F#

```
[<ExtensionAttribute>]
static member RelabelEdges :
    oldEdges : IEnumerable<IEdge> *
    graph : IGraph *
    newLabel : string -> unit
```

Parameters

oldEdges

Type: [System.Collections.Generic.IEnumerable\(IEdge\)](#)

the existing edges to "relabel"

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to add the new edge to

newLabel

Type: [System.String](#)

the label of the new edge

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable\(IEdge\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[EdgeHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.ElementHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.ElementHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class ElementHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class ElementHelpers
```

C++







```
[ExtensionAttribute]  
public ref class ElementHelpers abstract sealed
```
















F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type ElementHelpers = class end
```

The **ElementHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. |
|   | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. |
|   | GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. |

| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | HaveEqualIds | Simply tests if the element ids are equal(). |
|   | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). |
|   | RemoveProperties | Clear all the properties from an IEnumerable of elements. |
|   | RemoveProperty | Remove a property from all elements in the provided IEnumerable. |
|   | RenameProperty | Renames a property by removing the old key and adding the stored value to the new key. If property does not exist, nothing occurs. |
|   | SetProperties(IElement, IDictionary(String, Object)) | Set the properties of the provided element using the provided dictionary. |
|   | SetProperties(IElement, Object[]) | Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. |
|   | TypecastProperty | Typecasts a property value. This only works for casting to a class that has a constructor of the for new X(string). If no such constructor exists, an Exception is thrown and the original element property is left unchanged. |
|   | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.ElementHelpers Methods

The [ElementHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. |
|   | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. |
|   | GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. |
|   | HaveEqualIds | Simply tests if the element ids are equal(). |
|   | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). |
|   | RemoveProperties | Clear all the properties from an IEnumerable of elements. |
|   | RemoveProperty | Remove a property from all elements in the provided IEnumerable. |
|   | RenameProperty | Renames a property by removing the old key and adding the stored value to the new key. If property does not exist, nothing occurs. |
|   | SetProperties(IElement, IDictionary(String, Object)) | Set the properties of the provided element using the provided dictionary. |
|   | SetProperties(IElement, Object[]) | Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. |
|   | TypecastProperty | Typecasts a property value. This only works for casting to a class that has a constructor of the for new X(string). If no such constructor exists, an Exception is thrown and the original element property is left unchanged. |
|   | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided |

VelocityDB Class Library

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.AreEqual Method

A standard method for determining if two elements are equal. This method should be used by any `Element.equals()` implementation to ensure consistent behavior.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool AreEqual(  
    this IElement a,  
    Object b  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function AreEqual (  
    a As IElement,  
    b As Object  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool AreEqual(  
    IElement^ a,  
    Object^ b  
)
```

F#

```
[<ExtensionAttribute>]  
static member AreEqual :  
    a : IElement *  
    b : Object -> bool
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

The first element

b

Type: [System.Object](#)

The second element (as an object)

VelocityDB Class Library

Return Value

Type: [Boolean](#)

Whether the two elements are equal

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.CopyProperties Method

Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void CopyProperties(  
    this IElement from,  
    IElement to  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub CopyProperties (  
    from As IElement,  
    to As IElement  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void CopyProperties(  
    IElement^ from,  
    IElement^ to  
)
```

F#

```
[<ExtensionAttribute>]  
static member CopyProperties :  
    from : IElement *  
    to : IElement -> unit
```

Parameters

from

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
the element to copy properties from

to

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
the element to copy properties to

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.GetProperties Method

Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IDictionary<string, Object> GetProperties(  
    this IElement element  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function GetProperties (  
    element As IElement  
) As IDictionary(Of String, Object)
```

C++

```
public:  
[ExtensionAttribute]  
static IDictionary<String^, Object^>^ GetProperties(  
    IElement^ element  
)
```

F#

```
[<ExtensionAttribute>]  
static member GetProperties :  
    element : IElement -> IDictionary<string, Object>
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the element to get the properties of

Return Value

Type: [IDictionary\(String, Object\)](#)

a clone of the properties of the element

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#).

When you use instance method syntax to call this method, omit the first parameter. For more

information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

VelocityDB Class Library

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.HaveEqualIds Method

Simply tests if the element ids are equal().

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool HaveEqualIds(  
    this IElement a,  
    IElement b  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function HaveEqualIds (  
    a As IElement,  
    b As IElement  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool HaveEqualIds(  
    IElement^ a,  
    IElement^ b  
)
```

F#

```
[<ExtensionAttribute>]  
static member HaveEqualIds :  
    a : IElement *  
    b : IElement -> bool
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the first element

b

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the second element

Return Value

Type: [Boolean](#)

Whether the two elements have equal ids

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.HaveEqualProperties Method

Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals().

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool HaveEqualProperties (  
    this IElement a,  
    IElement b  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function HaveEqualProperties (  
    a As IElement,  
    b As IElement  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool HaveEqualProperties (  
    IElement^ a,  
    IElement^ b  
)
```

F#

```
[<ExtensionAttribute>]  
static member HaveEqualProperties :  
    a : IElement *  
    b : IElement -> bool
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
an element

b

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
an element

Return Value

Type: [Boolean](#)

whether the two elements have equal properties

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.RemoveProperties Method

Clear all the properties from an IEnumerable of elements.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void RemoveProperties (  
    this IEnumerable<IElement> elements  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub RemoveProperties (  
    elements As IEnumerable(Of IElement)  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void RemoveProperties (  
    IEnumerable<IElement^>^ elements  
)
```

F#

```
[<ExtensionAttribute>]  
static member RemoveProperties :  
    elements : IEnumerable<IElement> -> unit
```

Parameters

elements

Type: [System.Collections.Generic.IEnumerable\(IElement\)](#)

the elements to remove properties from

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable\(IElement\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.RemoveProperty Method

Remove a property from all elements in the provided IEnumerable.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void RemoveProperty(  
    this IEnumerable<IElement> elements,  
    string key  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub RemoveProperty (  
    elements As IEnumerable(Of IElement),  
    key As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void RemoveProperty(  
    IEnumerable<IElement^>^ elements,  
    String^ key  
)
```

F#

```
[<ExtensionAttribute>]  
static member RemoveProperty :  
    elements : IEnumerable<IElement> *  
    key : string -> unit
```

Parameters

elements

Type: [System.Collections.Generic.IEnumerable\(IElement\)](#)

the elements to remove the property from

key

Type: [System.String](#)

the property to remove by key

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable\(IElement\)](#). When you use instance method syntax to call this method, omit the first

VelocityDB Class Library

parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.RenameProperty Method

Renames a property by removing the old key and adding the stored value to the new key. If property does not exist, nothing occurs.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void RenameProperty(  
    this IEnumerable<IElement> elements,  
    string oldKey,  
    string newKey  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub RenameProperty (  
    elements As IEnumerable(Of IElement),  
    oldKey As String,  
    newKey As String  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void RenameProperty(  
    IEnumerable<IElement^>^ elements,  
    String^ oldKey,  
    String^ newKey  
)
```

F#

```
[<ExtensionAttribute>]  
static member RenameProperty :  
    elements : IEnumerable<IElement> *  
    oldKey : string *  
    newKey : string -> unit
```

Parameters

elements

Type: [System.Collections.Generic.IEnumerable\(IElement\)](#)

the elements to rename

oldKey

Type: [System.String](#)

the key to rename

newKey

Type: [System.String](#)
the key to rename to

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable\(IElement\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).





See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.SetProperties Method

Overload List

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | SetProperties(IElement, IDictionary<String, Object>) | Set the properties of the provided element using the provided dictionary. |
|   | SetProperties(IElement, Object[]) | Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. |

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.SetProperties Method (IElement, IDictionary(String, Object))

Set the properties of the provided element using the provided dictionary.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SetProperties (  
    this IElement element,  
    IDictionary<string, Object> properties  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SetProperties (  
    element As IElement,  
    properties As IDictionary(Of String, Object)  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SetProperties (  
    IElement^ element,  
    IDictionary<String^, Object^>^ properties  
)
```

F#

```
[<ExtensionAttribute>]  
static member SetProperties :  
    element : IElement *  
    properties : IDictionary<string, Object> -> unit
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the element to set the properties of

properties

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

the properties to set as a `IDictionary<string, object>`

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[SetProperties Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.SetProperties Method (IElement, Object[])

Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void SetProperties(  
    this IElement element,  
    params Object[] keysValues  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub SetProperties (  
    element As IElement,  
    ParamArray keysValues As Object()  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void SetProperties(  
    IElement^ element,  
    ... array<Object^>^ keysValues  
)
```

F#

```
[<ExtensionAttribute>]  
static member SetProperties :  
    element : IElement *  
    keysValues : Object[] -> unit
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
the element to set the properties of

keysValues

Type: [System.Object\[\]](#)

the key value pairs of the properties

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[SetProperties Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.TypecastProperty Method

Typecasts a property value. This only works for casting to a class that has a constructor of the form new X(string). If no such constructor exists, an Exception is thrown and the original element property is left unchanged.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void TypecastProperty(
    this IEnumerable<IElement> elements,
    string key,
    Type classCast
)
```

VB

```
<ExtensionAttribute>
Public Shared Sub TypecastProperty (
    elements As IEnumerable(Of IElement),
    key As String,
    classCast As Type
)
```

C++

```
public:
[ExtensionAttribute]
static void TypecastProperty(
    IEnumerable<IElement^>^ elements,
    String^ key,
    Type^ classCast
)
```

F#

```
[<ExtensionAttribute>]
static member TypecastProperty :
    elements : IEnumerable<IElement> *
    key : string *
    classCast : Type -> unit
```

Parameters

elements

Type: [System.Collections.Generic.IEnumerable<IElement>](#)
the elements to have their property typecasted

key

Type: [System.String](#)

the key for the property value to typecast

classCast

Type: [System.Type](#)

the class to typecast to

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEnumerable\(IElement\)](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ElementHelpers.ValidateProperty Method

Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateProperty(  
    this IElement element,  
    string key,  
    Object value  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub ValidateProperty (  
    element As IElement,  
    key As String,  
    value As Object  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void ValidateProperty(  
    IElement^ element,  
    String^ key,  
    Object^ value  
)
```

F#

```
[<ExtensionAttribute>]  
static member ValidateProperty :  
    element : IElement *  
    key : string *  
    value : Object -> unit
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)
the element for the property to be set

key

Type: [System.String](#)

VelocityDB Class Library

the key of the property

value

Type: [System.Object](#)

the value of the property

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IElement](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[ElementHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory Class

The ExceptionFactory provides standard exceptions that should be used by all Blueprints implementations. This ensures that the look-and-feel of all implementations are the same in terms of terminology and punctuation.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class ExceptionFactory
```

VB

```
Public NotInheritable Class ExceptionFactory
```

C++









```
public ref class ExceptionFactory abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type ExceptionFactory = class end
```

The **ExceptionFactory** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | BothIsNotSupported | |
|  | ClassIsNotIndexable | |
|  | EdgeIdCanNotBeNull | |
|  | EdgeWithIdAlreadyExist | |
|  | IndexAlreadyExists | |
|  | IndexDoesNotSupportClass | |
|  | PropertyKeyCanNotBeEmpty | |
|  | PropertyKeyCanNotBeNull | |
|  | PropertyKeyIdIsReserved | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------------------|--|
|  | PropertyKeyIsReserved | |
|  | PropertyKeyLabelsReservedForEdges | |
|  | PropertyValueCanNotBeNull | |
|  | TransactionAlreadyStarted | |
|  | VertexIdCanNotBeNull | |
|  | VertexWithIdAlreadyExists | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.ExceptionFactory Methods

The [ExceptionFactory](#) type exposes the following members.

Methods

| | Name | Description |
|--|---------------------------------------------------|-------------|
| | BothIsNotSupported | |
| | ClassIsNotIndexable | |
| | EdgeIdCanNotBeNull | |
| | EdgeWithIdAlreadyExist | |
| | IndexAlreadyExists | |
| | IndexDoesNotSupportClass | |
| | PropertyKeyCanNotBeEmpty | |
| | PropertyKeyCanNotBeNull | |
| | PropertyKeyIdsReserved | |
| | PropertyKeysReserved | |
| | PropertyKeyLabelsReservedForEdges | |
| | PropertyValueCanNotBeNull | |
| | TransactionAlreadyStarted | |
| | VertexIdCanNotBeNull | |
| | VertexWithIdAlreadyExists | |

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.BothIsNotSupported Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.BothIsNotSupported"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException BothIsNotSupported()
```

VB

```
Public Shared Function BothIsNotSupported As ArgumentException
```

C++

```
public:  
static ArgumentException^ BothIsNotSupported()
```

F#

```
static member BothIsNotSupported : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.BothIsNotSupported"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.ClassIsNotIndexable Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.ClassIsNotIndexable(System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException ClassIsNotIndexable(  
    Type clazz  
)
```

VB

```
Public Shared Function ClassIsNotIndexable (  
    clazz As Type  
) As ArgumentException
```

C++

```
public:  
static ArgumentException^ ClassIsNotIndexable(  
    Type^ clazz  
)
```

F#

```
static member ClassIsNotIndexable :  
    clazz : Type -> ArgumentException
```

Parameters

clazz

Type: [System.Type](#)

[Missing <param name="clazz"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.ClassIsNotIndexable(System.Type)"]

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.ClassIsNotIndexable(System.Type)"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.EdgeIdCanNotBeNull Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.EdgeIdCanNotBeNull"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException EdgeIdCanNotBeNull ()
```

VB

```
Public Shared Function EdgeIdCanNotBeNull As ArgumentException
```

C++

```
public:  
static ArgumentException^ EdgeIdCanNotBeNull ()
```

F#

```
static member EdgeIdCanNotBeNull : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.EdgeIdCanNotBeNull"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.EdgeWithIdAlreadyExist Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.EdgeWithIdAlreadyExist(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException EdgeWithIdAlreadyExist (
    Object id
)
```

VB

```
Public Shared Function EdgeWithIdAlreadyExist (
    id As Object
) As ArgumentException
```

C++

```
public:
static ArgumentException^ EdgeWithIdAlreadyExist (
    Object^ id
)
```

F#

```
static member EdgeWithIdAlreadyExist :
    id : Object -> ArgumentException
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.EdgeWithIdAlreadyExist(System.Object)"]

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.EdgeWithIdAlreadyExist(System.Object)"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.IndexAlreadyExists Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexAlreadyExists(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException IndexAlreadyExists(  
    string indexName  
)
```

VB

```
Public Shared Function IndexAlreadyExists (  
    indexName As String  
) As ArgumentException
```

C++

```
public:  
static ArgumentException^ IndexAlreadyExists(  
    String^ indexName  
)
```

F#

```
static member IndexAlreadyExists :  
    indexName : string -> ArgumentException
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexAlreadyExists(System.String)"]

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexAlreadyExists(System.String)"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.IndexDoesNotSupportClass Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexDoesNotSupportClass(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static InvalidOperationException IndexDoesNotSupportClass (
    string indexName,
    Type clazz
)
```

VB

```
Public Shared Function IndexDoesNotSupportClass (
    indexName As String,
    clazz As Type
) As InvalidOperationException
```

C++

```
public:
static InvalidOperationException^ IndexDoesNotSupportClass (
    String^ indexName,
    Type^ clazz
)
```

F#

```
static member IndexDoesNotSupportClass :
    indexName : string *
    clazz : Type -> InvalidOperationException
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexDoesNotSupportClass(System.String,System.Type)"]

clazz

Type: [System.Type](#)

[Missing <param name="clazz"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexDoesNotSupportClass(System.String,System.Type)"]

Return Value

Type: [InvalidOperationException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.IndexDoesNotSupportClass(System.String,System.Type)"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyKeyCanNotBeEmpty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyCanNotBeEmpty"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyKeyCanNotBeEmpty()
```

VB

```
Public Shared Function PropertyKeyCanNotBeEmpty As ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyKeyCanNotBeEmpty()
```

F#

```
static member PropertyKeyCanNotBeEmpty : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyCanNotBeEmpty"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyKeyCanNotBeNull Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyCanNotBeNull"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyKeyCanNotBeNull ()
```

VB

```
Public Shared Function PropertyKeyCanNotBeNull As ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyKeyCanNotBeNull ()
```

F#

```
static member PropertyKeyCanNotBeNull : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyCanNotBeNull"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyKeyIdsReserved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyIdsReserved"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyKeyIdIsReserved()
```

VB

```
Public Shared Function PropertyKeyIdIsReserved As ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyKeyIdIsReserved()
```

F#

```
static member PropertyKeyIdIsReserved : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyIdsReserved"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyKeysReserved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeysReserved(System.String)"
]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyKeyIsReserved(  
    string key  
)
```

VB

```
Public Shared Function PropertyKeyIsReserved (  
    key As String  
) As ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyKeyIsReserved(  
    String^ key  
)
```

F#

```
static member PropertyKeyIsReserved :  
    key : string -> ArgumentException
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeysReserved(System.String)"
]

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeysReserved(System.String)"
]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyKeyLabelsReservedForEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyLabelsReservedForEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyKeyLabelIsReservedForEdges ()
```

VB

```
Public Shared Function PropertyKeyLabelIsReservedForEdges As  
ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyKeyLabelIsReservedForEdges ()
```

F#

```
static member PropertyKeyLabelIsReservedForEdges : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyKeyLabelsReservedForEdges"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.PropertyValueCanNotBeNull Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyValueCanNotBeNull"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException PropertyValueCanNotBeNull ()
```

VB

```
Public Shared Function PropertyValueCanNotBeNull As ArgumentException
```

C++

```
public:  
static ArgumentException^ PropertyValueCanNotBeNull ()
```

F#

```
static member PropertyValueCanNotBeNull : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.PropertyValueCanNotBeNull"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.TransactionAlreadyStarted Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.TransactionAlreadyStarted"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static InvalidOperationException TransactionAlreadyStarted()
```

VB

```
Public Shared Function TransactionAlreadyStarted As InvalidOperationException
```

C++

```
public:  
static InvalidOperationException^ TransactionAlreadyStarted()
```

F#

```
static member TransactionAlreadyStarted : unit -> InvalidOperationException
```

Return Value

Type: [InvalidOperationException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.TransactionAlreadyStarted"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.VertexIdCanNotBeNull Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.VertexIdCanNotBeNull"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException VertexIdCanNotBeNull ()
```

VB

```
Public Shared Function VertexIdCanNotBeNull As ArgumentException
```

C++

```
public:  
static ArgumentException^ VertexIdCanNotBeNull ()
```

F#

```
static member VertexIdCanNotBeNull : unit -> ArgumentException
```

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.VertexIdCanNotBeNull"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

ExceptionFactory.VertexWithIdAlreadyExists Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.VertexWithIdAlreadyExists(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ArgumentException VertexWithIdAlreadyExists (
    Object id
)
```

VB

```
Public Shared Function VertexWithIdAlreadyExists (
    id As Object
) As ArgumentException
```

C++

```
public:
static ArgumentException^ VertexWithIdAlreadyExists (
    Object^ id
)
```

F#

```
static member VertexWithIdAlreadyExists :
    id : Object -> ArgumentException
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.VertexWithIdAlreadyExists(System.Object)"]

Return Value

Type: [ArgumentException](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.ExceptionFactory.VertexWithIdAlreadyExists(System.Object)"]

See Also

[ExceptionFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

GraphHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.GraphHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.GraphHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class GraphHelpers
```

C++







```
[ExtensionAttribute]  
public ref class GraphHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type GraphHelpers = class end
```

The **GraphHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AddEdge | Add an edge to the graph with specified id and provided properties. |
|   | AddVertex | Add a vertex to the graph with specified id and provided properties. |
|   | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. |







See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

GraphHelpers.GraphHelpers Methods

The [GraphHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AddEdge | Add an edge to the graph with specified id and provided properties. |
|   | AddVertex | Add a vertex to the graph with specified id and provided properties. |
|   | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. |

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

GraphHelpers.AddEdge Method

Add an edge to the graph with specified id and provided properties.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEdge AddEdge(
    this IGraph graph,
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label,
    params Object[] properties
)
```

VB

```
<ExtensionAttribute>
Public Shared Function AddEdge (
    graph As IGraph,
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String,
    ParamArray properties As Object()
) As IEdge
```

C++

```
public:
[ExtensionAttribute]
static IEdge^ AddEdge(
    IGraph^ graph,
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label,
    ... array<Object^>^ properties
)
```

F#

```
[<ExtensionAttribute>]
static member AddEdge :
    graph : IGraph *
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string *
    properties : Object[] -> IEdge
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to create the edge in

id

Type: [System.Object](#)

the id of the edge to create

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the outgoing/tail vertex of the edge

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the incoming/head vertex of the edge

label

Type: [System.String](#)

the label of the edge

properties

Type: [System.Object\[\]](#)

the properties of the edge to add (must be string,object,string,object,...)

Return Value

Type: [IEdge](#)

the edge created in the graph with the provided properties set

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

GraphHelpers.AddVertex Method

Add a vertex to the graph with specified id and provided properties.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertex AddVertex(  
    this IGraph graph,  
    Object id,  
    params Object[] properties  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function AddVertex (  
    graph As IGraph,  
    id As Object,  
    ParamArray properties As Object()  
) As IVertex
```

C++

```
public:  
[ExtensionAttribute]  
static IVertex^ AddVertex(  
    IGraph^ graph,  
    Object^ id,  
    ... array<Object^>^ properties  
)
```

F#

```
[<ExtensionAttribute>]  
static member AddVertex :  
    graph : IGraph *  
    id : Object *  
    properties : Object[] -> IVertex
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to create a vertex in

id

Type: [System.Object](#)

the id of the vertex to create

properties

Type: [System.Object\[\]](#)

the properties of the vertex to add (must be string,object,string,object,...)

Return Value

Type: [IVertex](#)

the vertex created in the graph with the provided properties set

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

GraphHelpers.CopyGraph Method

Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void CopyGraph(  
    this IGraph from,  
    IGraph to  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Sub CopyGraph (  
    from As IGraph,  
    to As IGraph  
)
```

C++

```
public:  
[ExtensionAttribute]  
static void CopyGraph(  
    IGraph^ from,  
    IGraph^ to  
)
```

F#

```
[<ExtensionAttribute>]  
static member CopyGraph :  
    from : IGraph *  
    to : IGraph -> unit
```

Parameters

from

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to copy from

to

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to copy to

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[GraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

IndexableGraphHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.IndexableGraphHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IndexableGraphHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class IndexableGraphHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class IndexableGraphHelpers
```

C++


```
[ExtensionAttribute]  
public ref class IndexableGraphHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type IndexableGraphHelpers = class end
```

The **IndexableGraphHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. |



See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

IndexableGraphHelpers.IndexableGraphHelpers Methods

The [IndexableGraphHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. |

See Also

[IndexableGraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

IndexableGraphHelpers.AddUniqueVertex Method

Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertex AddUniqueVertex(
    this IIndexableGraph graph,
    Object id,
    IIndex index,
    string uniqueKey,
    Object uniqueValue
)
```

VB

```
<ExtensionAttribute>
Public Shared Function AddUniqueVertex (
    graph As IIndexableGraph,
    id As Object,
    index As IIndex,
    uniqueKey As String,
    uniqueValue As Object
) As IVertex
```

C++

```
public:
[ExtensionAttribute]
static IVertex^ AddUniqueVertex(
    IIndexableGraph^ graph,
    Object^ id,
    IIndex^ index,
    String^ uniqueKey,
    Object^ uniqueValue
)
```

F#

```
[<ExtensionAttribute>]
static member AddUniqueVertex :
    graph : IIndexableGraph *
    id : Object *
    index : IIndex *
    uniqueKey : string *
    uniqueValue : Object -> IVertex
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

the graph to add the vertex to

id

Type: [System.Object](#)

the id of the vertex to create (can be null)

index

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

the index to determine if another vertex with the same key/value exists

uniqueKey

Type: [System.String](#)

the key to check on for uniqueness of the vertex

uniqueValue

Type: [System.Object](#)

the value to check on for uniqueness of the vertex

Return Value

Type: [IVertex](#)

the newly created vertex or the vertex that satisfies the uniqueness criteria

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IIndexableGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[IndexableGraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

KeyIndexableGraphHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.KeyIndexableGraphHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.KeyIndexableGraphHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class KeyIndexableGraphHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class KeyIndexableGraphHelpers
```

C++


```
[ExtensionAttribute]  
public ref class KeyIndexableGraphHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type KeyIndexableGraphHelpers = class end
```

The **KeyIndexableGraphHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. |



See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

KeyIndexableGraphHelpers.KeyIndexableGraphHelpers Methods

The [KeyIndexableGraphHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. |

See Also

[KeyIndexableGraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

KeyIndexableGraphHelpers.ReIndexElements(T) Method

For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static long ReIndexElements<T>(
    this IGraph graph,
    IEnumerable<T> elements,
    IEnumerable<string> keys
)
where T : IElement
```

VB

```
<ExtensionAttribute>
Public Shared Function ReIndexElements(Of T As IElement) (
    graph As IGraph,
    elements As IEnumerable(Of T),
    keys As IEnumerable(Of String)
) As Long
```

C++

```
public:
[ExtensionAttribute]
generic<typename T>
where T : IElement
static long long ReIndexElements(
    IGraph^ graph,
    IEnumerable<T>^ elements,
    IEnumerable<String^>^ keys
)
```

F#

```
[<ExtensionAttribute>]
static member ReIndexElements :
    graph : IGraph *
    elements : IEnumerable<'T> *
    keys : IEnumerable<string> -> int64 when 'T : IElement
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph containing the provided elements

elements

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

the elements to index into the key indices

keys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the key indices

Type Parameters

T

[Missing <typeparam name="T" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.KeyIndexableGraphHelpers.ReIndexElements`1(VelocityGraph.Frontenac.Blueprints.IGraph,System.Collections.Generic.IEnumerable{`0},System.Collections.Generic.IEnumerable{System.String})"]

Return Value

Type: [Int64](#)

the number of element properties that were indexed

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[KeyIndexableGraphHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

Multiterable(TS) Class

A helper class that is used to combine multiple iterables into a single closeable IEnumerable.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Multiterable(TS)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class MultiIterable<TS> : ICloseableIterable<TS>,
    IDisposable, IEnumerable<TS>, IEnumerable
```

VB

```
Public Class MultiIterable(Of TS)
    Implements ICloseableIterable(Of TS), IDisposable, IEnumerable(Of TS),
    IEnumerable
```

C++

```
generic<typename TS>
public ref class MultiIterable : ICloseableIterable<TS>,
    IDisposable, IEnumerable<TS>, IEnumerable
```

F#

```
type MultiIterable<'TS> =
    class
        interface ICloseableIterable<'TS>
        interface IDisposable
        interface IEnumerable<'TS>
        interface IEnumerable
    end
```


Type Parameters

TS



[Missing <typeparam name="TS"/> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Multiterable`1"]

The Multiterable(TS) type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|----------------------------------------------------------|
|  | Multiterable(TS) | Initializes a new instance of the Multiterable(TS) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------|
|  | Dispose | Releases all resources used by the Multiterable(TS) |
|  | GetEnumerator | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

Multiterable(TS) Constructor

Initializes a new instance of the [Multiterable\(TS\)](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public MultiIterable(  
    IEnumerable<IEnumerable<TS>> iterables  
)
```

VB

```
Public Sub New (  
    iterables As IEnumerable(Of IEnumerable(Of TS))  
)
```

C++

```
public:  
MultiIterable(  
    IEnumerable<IEnumerable<TS>^>^ iterables  
)
```

F#

```
new :  
    iterables : IEnumerable<IEnumerable<'TS>> -> MultiIterable
```

Parameters

iterables

Type: [System.Collections.Generic.IEnumerable\(IEnumerable\(TS\)\)](#)

[Missing <param name="iterables"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Multiterable`1.#ctor(System.Collections.Generic.IEnumerable{System.Collections.Generic.IEnumerable{`0}})"]

See Also

[Multiterable\(TS\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

Multiterable(TS).Multiterable(TS) Methods

The [Multiterable\(TS\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | Dispose | |
|  | GetEnumerator | |

See Also

[Multiterable\(TS\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

Multiterable(TS).Dispose Method

Releases all resources used by the [Multiterable\(TS\)](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[Multiterable\(TS\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

Multiliterable(TS).GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Multiliterable`1.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<TS> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of TS)
```

C++

```
public:  
virtual IEnumerator<TS>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<'TS>  
override GetEnumerator : unit -> IEnumerator<'TS>
```

Return Value

Type: [IEnumerator\(TS\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Multiliterable`1.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[Multiliterable\(TS\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

PropertyFilteredIterable(T) Class

This is a helper class for filtering an IEnumerable of elements by their key/value. Useful for Graph implementations that do not support automatic key indices and need to filter on Graph.getVertices/Edges(key,value).

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable(T)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PropertyFilteredIterable<T> : ICloseableIterable<T>,
    IDisposable, IEnumerable<T>, IEnumerable
where T : class, IElement
```

VB

```
Public Class PropertyFilteredIterable(Of T As {Class, IElement})
    Implements ICloseableIterable(Of T), IDisposable, IEnumerable(Of T),
    IEnumerable
```

C++

```
generic<typename T>
where T : ref class, IElement
public ref class PropertyFilteredIterable : ICloseableIterable<T>,
    IDisposable, IEnumerable<T>, IEnumerable
```

F#

```
type PropertyFilteredIterable<'T when 'T : not struct and IElement> =
    class
        interface ICloseableIterable<'T>
        interface IDisposable
        interface IEnumerable<'T>
        interface IEnumerable
    end
```


Type Parameters

T



[Missing <typeparam name="T"/> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1"]

The PropertyFilteredIterable(T) type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|---------------------------------------------------------------------|
|  | PropertyFilteredIterable(T) | Initializes a new instance of the PropertyFilteredIterable(T) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------|
|  | Dispose | Releases all resources used by the PropertyFilteredIterable(T) |
|  | GetEnumerator | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

PropertyFilteredIterable(T) Constructor

Initializes a new instance of the [PropertyFilteredIterable\(T\)](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PropertyFilteredIterable(  
    string key,  
    Object value,  
    IEnumerable<T> iterable  
)
```

VB

```
Public Sub New (  
    key As String,  
    value As Object,  
    iterable As IEnumerable(Of T)  
)
```

C++

```
public:  
PropertyFilteredIterable(  
    String^ key,  
    Object^ value,  
    IEnumerable<T>^ iterable  
)
```

F#

```
new :  
    key : string *  
    value : Object *  
    iterable : IEnumerable<'T> -> PropertyFilteredIterable
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1.#ctor(System.String,System.Object,System.Collections.Generic.IEnumerable{`0})"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1.#ctor(System.String,System.Object,System.Collections.Generic.IEnumerable{`0})"]

iterable

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="iterable"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1.#ctor(System.String,System.Object,System.Collections.Generic.IEnumerable{`0})"]

See Also

[PropertyFilteredIterable\(T\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

PropertyFilteredIterable(T).PropertyFilteredIterable(T) Methods

The [PropertyFilteredIterable\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | Dispose | |
|  | GetEnumerator | |

See Also

[PropertyFilteredIterable\(T\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

PropertyFilteredIterable(T).Dispose Method

Releases all resources used by the [PropertyFilteredIterable\(T\)](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[PropertyFilteredIterable\(T\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

PropertyFilteredIterable(T).GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<T> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of T)
```

C++

```
public:  
virtual IEnumerator<T>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<'T>  
override GetEnumerator : unit -> IEnumerator<'T>
```

Return Value

Type: [IEnumerator\(T\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.PropertyFilteredIterable`1.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[PropertyFilteredIterable\(T\) Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory Class

A collection of helpful methods for creating standard toString() representations of graph-related objects.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.StringFactory

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class StringFactory
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class StringFactory
```

C++





```
[ExtensionAttribute]  
public ref class StringFactory abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type StringFactory = class end
```

The **StringFactory** type exposes the following members.

















Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | EdgeString | |
|  | GraphString | |
|  | IndexString | |
|  | VertexString | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------|-------------|
|  | Arrow | |
|  | Colon | |

VelocityDB Class Library

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--|
|   | Dash | |
|   | E | |
|   | EmptyString | |
|   | Id | |
|   | Label | |
|   | LBracket | |
|   | RBracket | |
|   | V | |





See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.StringFactory Methods

The [StringFactory](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | EdgeString | |
|  | GraphString | |
|  | IndexString | |
|  | VertexString | |

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.EdgeString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.EdgeString(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string EdgeString(  
    this IEdge edge  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function EdgeString (  
    edge As IEdge  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ EdgeString(  
    IEdge^ edge  
)
```

F#

```
[<ExtensionAttribute>]  
static member EdgeString :  
    edge : IEdge -> string
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.EdgeString(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.EdgeString(VelocityGraph.Frontenac.Blueprints.IEdge)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IEdge](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.GraphString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.GraphString(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string GraphString(  
    this IGraph graph,  
    string internalString  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function GraphString (  
    graph As IGraph,  
    internalString As String  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ GraphString(  
    IGraph^ graph,  
    String^ internalString  
)
```

F#

```
[<ExtensionAttribute>]  
static member GraphString :  
    graph : IGraph *  
    internalString : string -> string
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.GraphString(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

internalString

Type: [System.String](#)

[Missing <param name="internalString"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.GraphString(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.GraphString(VelocityGraph.Frontenac.Blueprints.IGraph,System.String)"]

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IGraph](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.IndexString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.IndexString(VelocityGraph.Frontenac.Blueprints.IIndex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string IndexString(  
    this IIndex index  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function IndexString (  
    index As IIndex  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ IndexString(  
    IIndex^ index  
)
```

F#

```
[<ExtensionAttribute>]  
static member IndexString :  
    index : IIndex -> string
```

Parameters

index

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="index"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.IndexString(VelocityGraph.Frontenac.Blueprints.IIndex)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.IndexString(VelocityGraph.Frontenac.Blueprints.IIndex)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IIndex](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.VertexString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.VertexString(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static string VertexString(  
    this IVertex vertex  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function VertexString (  
    vertex As IVertex  
) As String
```

C++

```
public:  
[ExtensionAttribute]  
static String^ VertexString(  
    IVertex^ vertex  
)
```

F#

```
[<ExtensionAttribute>]  
static member VertexString :  
    vertex : IVertex -> string
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.VertexString(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.VertexString(VelocityGraph.Frontenac.Blueprints.IVertex)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IVertex](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also




















[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.StringFactory Fields

The [StringFactory](#) type exposes the following members.

Fields

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-------------|
|   | Arrow | |
|   | Colon | |
|   | Dash | |
|   | E | |
|   | EmptyString | |
|   | Id | |
|   | Label | |
|   | LBracket | |
|   | RBracket | |
|   | V | |

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.Arrow Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.Arrow"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Arrow = "->"
```

VB

```
Public Const Arrow As String = "->"
```

C++

```
public:  
literal String^ Arrow = "->"
```

F#

```
static val mutable Arrow: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.Colon Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.Colon"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Colon = ":"
```

VB

```
Public Const Colon As String = ":"
```

C++

```
public:  
literal String^ Colon = ":"
```

F#

```
static val mutable Colon: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.Dash Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.Dash"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Dash = "-"
```

VB

```
Public Const Dash As String = "-"
```

C++

```
public:  
literal String^ Dash = "-"
```

F#

```
static val mutable Dash: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.E Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.E"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string E = "e"
```

VB

```
Public Const E As String = "e"
```

C++

```
public:  
literal String^ E = "e"
```

F#

```
static val mutable E: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.EmptyString Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.EmptyString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string EmptyString = ""
```

VB

```
Public Const EmptyString As String = ""
```

C++

```
public:  
literal String^ EmptyString = ""
```

F#

```
static val mutable EmptyString: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.Id Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Id = "id"
```

VB

```
Public Const Id As String = "id"
```

C++

```
public:  
literal String^ Id = "id"
```

F#

```
static val mutable Id: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.Label Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Label = "label"
```

VB

```
Public Const Label As String = "label"
```

C++

```
public:  
literal String^ Label = "label"
```

F#

```
static val mutable Label: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.LBracket Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.LBracket"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string LBracket = "["
```

VB

```
Public Const LBracket As String = "["
```

C++

```
public:  
literal String^ LBracket = "["
```

F#

```
static val mutable LBracket: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.RBracket Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.RBracket"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string RBracket = "]"
```

VB

```
Public Const RBracket As String = "]"
```

C++

```
public:  
literal String^ RBracket = "]"
```

F#

```
static val mutable RBracket: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

StringFactory.V Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.StringFactory.V"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string V = "v"
```

VB

```
Public Const V As String = "v"
```

C++

```
public:  
literal String^ V = "v"
```

F#

```
static val mutable V: string
```

Field Value

Type: [String](#)

See Also

[StringFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VertexHelpers Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.VertexHelpers"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.VertexHelpers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class VertexHelpers
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class VertexHelpers
```

C++



```
[ExtensionAttribute]  
public ref class VertexHelpers abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type VertexHelpers = class end
```

The **VertexHelpers** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------|
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. |



See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VertexHelpers.VertexHelpers Methods

The [VertexHelpers](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|--------------------------------------------------------------------|
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. |

See Also

[VertexHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VertexHelpers.HaveEqualEdges Method

Test whether the two vertices have equal edge sets

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool HaveEqualEdges (  
    this IVertex a,  
    IVertex b,  
    bool checkIdEquality  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function HaveEqualEdges (  
    a As IVertex,  
    b As IVertex,  
    checkIdEquality As Boolean  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool HaveEqualEdges (  
    IVertex^ a,  
    IVertex^ b,  
    bool checkIdEquality  
)
```

F#

```
[<ExtensionAttribute>]  
static member HaveEqualEdges :  
    a : IVertex *  
    b : IVertex *  
    checkIdEquality : bool -> bool
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the first vertex

b

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the second vertex

VelocityDB Class Library

checkIdEquality

Type: [System.Boolean](#)

whether to check on vertex and edge ids

Return Value

Type: [Boolean](#)

whether the two vertices have the same edge sets

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IVertex](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[VertexHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VertexHelpers.HaveEqualNeighborhood Method

Test whether the two vertices have equal properties and edge sets.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static bool HaveEqualNeighborhood(  
    this IVertex a,  
    IVertex b,  
    bool checkIdEquality  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function HaveEqualNeighborhood (  
    a As IVertex,  
    b As IVertex,  
    checkIdEquality As Boolean  
) As Boolean
```

C++

```
public:  
[ExtensionAttribute]  
static bool HaveEqualNeighborhood(  
    IVertex^ a,  
    IVertex^ b,  
    bool checkIdEquality  
)
```

F#

```
[<ExtensionAttribute>]  
static member HaveEqualNeighborhood :  
    a : IVertex *  
    b : IVertex *  
    checkIdEquality : bool -> bool
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the first vertex

b

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the second vertex

VelocityDB Class Library

checkIdEquality

Type: [System.Boolean](#)

whether to check on vertex and edge ids

Return Value

Type: [Boolean](#)

whether the two vertices are semantically the same

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [IVertex](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[VertexHelpers Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VerticesFromEdgesIterable Class

VerticesFromEdgesIterable is a helper class that returns vertices that meet the direction/label criteria of the incident edges.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class VerticesFromEdgesIterable : IEnumerable<IVertex>,
    IEnumerable
```

VB

```
Public Class VerticesFromEdgesIterable
    Implements IEnumerable(Of IVertex), IEnumerable
```

C++


```
public ref class VerticesFromEdgesIterable : IEnumerable<IVertex^>,
    IEnumerable
```

F#


```
type VerticesFromEdgesIterable =
    class
        interface IEnumerable<IVertex>
        interface IEnumerable
    end
```

The **VerticesFromEdgesIterable** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------|
|  | VerticesFromEdgesIterable | Initializes a new instance of the VerticesFromEdgesIterable class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-------------|
|  | GetEnumerator | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VerticesFromEdgesIterable Constructor

Initializes a new instance of the [VerticesFromEdgesIterable](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VerticesFromEdgesIterable(  
    IVertex vertex,  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    direction As Direction,  
    ParamArray labels As String()  
)
```

C++

```
public:  
VerticesFromEdgesIterable(  
    IVertex^ vertex,  
    Direction direction,  
    ... array<String^>^ labels  
)
```

F#

```
new :  
    vertex : IVertex *  
    direction : Direction *  
    labels : string[] -> VerticesFromEdgesIterable
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

See Also


[VerticesFromEdgesIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VerticesFromEdgesIterable.VerticesFromEdgesIterable Methods

The [VerticesFromEdgesIterable](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | GetEnumerator | |

See Also

[VerticesFromEdgesIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VerticesFromEdgesIterable.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<IVertex> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of IVertex)
```

C++

```
public:  
virtual IEnumerator<IVertex^> GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<IVertex>  
override GetEnumerator : unit -> IEnumerator<IVertex>
```

Return Value

Type: [IEnumerator<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.VerticesFromEdgesIterable.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[VerticesFromEdgesIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T) Class

[Missing <summary> documentation for
"T:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable(T)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappingCloseableIterable<T> : ICloseableIterable<T>,
    IDisposable, IEnumerable<T>, IEnumerable
```

VB

```
Public Class WrappingCloseableIterable(Of T)
    Implements ICloseableIterable(Of T), IDisposable, IEnumerable(Of T),
    IEnumerable
```

C++

```
generic<typename T>
public ref class WrappingCloseableIterable : ICloseableIterable<T>,
    IDisposable, IEnumerable<T>, IEnumerable
```

F#

```
type WrappingCloseableIterable<'T> =
    class
        interface ICloseableIterable<'T>
        interface IDisposable
        interface IEnumerable<'T>
        interface IEnumerable
    end
```


Type Parameters

T




[Missing <typeparam name="T"/> documentation for
"T:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1"]

The WrappingCloseableIterable(T) type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------|----------------------------------------------------------------------|
|  | WrappingCloseableIterable(T) | Initializes a new instance of the WrappingCloseableIterable(T) class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------|
|  | Dispose | Releases all resources used by the WrappingCloseableIterable(T) |
|  | GetEnumerator | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T) Constructor

Initializes a new instance of the [WrappingCloseableIterable\(T\)](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappingCloseableIterable(  
    IEnumerable<T> iterable  
)
```

VB

```
Public Sub New (  
    iterable As IEnumerable(Of T)  
)
```

C++

```
public:  
WrappingCloseableIterable(  
    IEnumerable<T>^ iterable  
)
```

F#

```
new :  
    iterable : IEnumerable<'T> -> WrappingCloseableIterable
```

Parameters

iterable

Type: [System.Collections.Generic.IEnumerable\(T\)](#)

[Missing <param name="iterable"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1.#ctor(System.Collections.Generic.IEnumerable{`0})"]

See Also




[WrappingCloseableIterable\(T\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T).WrappingCloseableIterable(T) Methods

The [WrappingCloseableIterable\(T\)](#) generic type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------|
|  | Dispose | |
|  | GetEnumerator | |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[WrappingCloseableIterable\(T\) Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T).Dispose Method

Releases all resources used by the [WrappingCloseableIterable\(T\)](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose() sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[WrappingCloseableIterable\(T\)Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T).GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<T> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of T)
```

C++

```
public:  
virtual IEnumerable<T>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<'T>  
override GetEnumerator : unit -> IEnumerable<'T>
```

Return Value

Type: [IEnumerable\(T\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[WrappingCloseableIterable\(T\) Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

WrappingCloseableIterable(T).ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.WrappingCloseableIterable`1.ToString"]

See Also



[WrappingCloseableIterable\(T\) Class](#)

[VelocityGraph.Frontenac.Blueprints.Util Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.IO Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.IO"]

Classes

| Class | Description |
|------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------|
|  LexicographicalElementComparator | Elements are sorted in lexicographical order of IDs. |
|  StreamTokenizer | |

LexicographicalElementComparator Class

Elements are sorted in lexicographical order of IDs.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.LexicographicalElementComparator

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class LexicographicalElementComparator : IComparer<IElement>
```

VB

```
Public Class LexicographicalElementComparator
    Implements IComparer(Of IElement)
```

C++


```
public ref class LexicographicalElementComparator : IComparer<IElement^>
```

F#


```
type LexicographicalElementComparator =
    class
        interface IComparer<IElement>
    end
```

The **LexicographicalElementComparator** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------|---------------------------------------------------------------------------------|
|  | LexicographicalElementComparator | Initializes a new instance of the LexicographicalElementComparator class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

LexicographicalElementComparator Constructor

Initializes a new instance of the [LexicographicalElementComparator](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public LexicographicalElementComparator ()
```

VB

```
Public Sub New
```

C++

```
public:  
LexicographicalElementComparator ()
```

F#

```
new : unit -> LexicographicalElementComparator
```

See Also


[LexicographicalElementComparator Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

LexicographicalElementComparator.LexicographicalElementComparator Methods

The [LexicographicalElementComparator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|-------------|
|  | Compare | |

See Also

[LexicographicalElementComparator Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

LexicographicalElementComparator.Compare Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.LexicographicalElementComparator.Compare(VelocityGraph.Frontenac.Blueprints.IElement,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Compare(  
    IElement a,  
    IElement b  
)
```

VB

```
Public Function Compare (  
    a As IElement,  
    b As IElement  
) As Integer
```

C++

```
public:  
virtual int Compare(  
    IElement^ a,  
    IElement^ b  
) sealed
```

F#

```
abstract Compare :  
    a : IElement *  
    b : IElement -> int  
override Compare :  
    a : IElement *  
    b : IElement -> int
```

Parameters

a

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="a"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.LexicographicalElementComparator.Compare(VelocityGraph.Frontenac.Blueprints.IElement,VelocityGraph.Frontenac.Blueprints.IElement)"]

b

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="b"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.LexicographicalElementComparator.Compare(VelocityGraph.Frontenac.Blueprints.IElement,VelocityGraph.Frontenac.Blueprints.IElement)"]

Return Value

Type: [Int32](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.LexicographicalElementComparator.Compare(VelocityGraph.Frontenac.Blueprints.IElement,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IComparer\(T\).Compare\(T, T\)](#)

See Also

[LexicographicalElementComparator Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class StreamTokenizer : IEnumerable<int>,
    IEnumerable
```

VB

```
Public Class StreamTokenizer
    Implements IEnumerable(Of Integer), IEnumerable
```

C++


```
public ref class StreamTokenizer : IEnumerable<int>,
    IEnumerable
```

F#




```
type StreamTokenizer =
    class
        interface IEnumerable<int>
        interface IEnumerable
    end
```




The **StreamTokenizer** type exposes the following members.

Constructors














| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|----------------------------------------------------------------|
|  | StreamTokenizer | Initializes a new instance of the StreamTokenizer class |

Properties






| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-------------|
|  | LineNumber | |
|  | LowerCaseMode | |
|  | NumberValue | |

| | | |
|-----------------------------------------------------------------------------------|------------------------------------|--|
|  | SlashSlashComments | |
|  | SlashStarComments | |
|  | StringValue | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------------------------------------------|
|  | CommentChar | |
|  | EollsSignificant | |
|  | GetEnumerator | |
|  | NextToken | |
|  | OrdinaryChar | |
|  | OrdinaryChars | |
|  | ParseNumbers | |
|  | PushBack | |
|  | QuoteChar | |
|  | ResetSyntax | |
|  | ToString | (Overrides Object.ToString() .) |
|  | WhitespaceChars | |
|  | WordChars | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | TtEof | |
|  | TtEol | |
|  | TtNumber | |
|  | TtWord | |
|  | Ttype | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer Constructor

Initializes a new instance of the [StreamTokenizer](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public StreamTokenizer(  
    StreamReader r  
)
```

VB

```
Public Sub New (  
    r As StreamReader  
)
```

C++

```
public:  
StreamTokenizer(  
    StreamReader^ r  
)
```

F#

```
new :  
    r : StreamReader -> StreamTokenizer
```

Parameters

r

Type: [System.IO.StreamReader](#)

[Missing <param name="r"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.#ctor(System.IO.StreamReader)"]

See Also







[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.StreamTokenizer Properties

The [StreamTokenizer](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------|
|  | LineNumber | |
|  | LowerCaseMode | |
|  | NumberValue | |
|  | SlashSlashComments | |
|  | SlashStarComments | |
|  | StringValue | |

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.LineNumber Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.LineNumber"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int LineNumber { get; }
```

VB

```
Public ReadOnly Property LineNumber As Integer  
    Get
```

C++

```
public:  
property int LineNumber {  
    int get ();  
}
```

F#

```
member LineNumber : int with get
```

Property Value

Type: [Int32](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.LowerCaseMode Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.LowerCaseMode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool LowerCaseMode { set; }
```

VB

```
Public WriteOnly Property LowerCaseMode As Boolean  
    Set
```

C++

```
public:  
property bool LowerCaseMode {  
    void set (bool value);  
}
```

F#

```
member LowerCaseMode : bool with set
```

Property Value

Type: [Boolean](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.NumberValue Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.NumberValue"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public double NumberValue { get; }
```

VB

```
Public ReadOnly Property NumberValue As Double  
    Get
```

C++

```
public:  
property double NumberValue {  
    double get ();  
}
```

F#

```
member NumberValue : float with get
```

Property Value

Type: [Double](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.SlashSlashComments Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.SlashSlashComments"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SlashSlashComments { get; set; }
```

VB

```
Public Property SlashSlashComments As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SlashSlashComments {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SlashSlashComments : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.SlashStarComments Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.SlashStarComments"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool SlashStarComments { get; set; }
```

VB

```
Public Property SlashStarComments As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool SlashStarComments {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member SlashStarComments : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.StringValue Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.StringValue"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string StringValue { get; }
```

VB

```
Public ReadOnly Property StringValue As String  
    Get
```

C++

```
public:  
property String^ StringValue {  
    String^ get ();  
}
```

F#

```
member StringValue : string with get
```

Property Value

Type: [String](#)

See Also














[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.StreamTokenizer Methods

The [StreamTokenizer](#) type exposes the following members.

Methods

| | Name | Description |
|------------------------------------------------------------------------------------|----------------------------------|-------------------------------------------------|
|  | CommentChar | |
|  | EolIsSignificant | |
|  | GetEnumerator | |
|  | NextToken | |
|  | OrdinaryChar | |
|  | OrdinaryChars | |
|  | ParseNumbers | |
|  | PushBack | |
|  | QuoteChar | |
|  | ResetSyntax | |
|  | ToString | (Overrides Object.ToString() .) |
|  | WhitespaceChars | |
|  | WordChars | |

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.CommentChar Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.CommentChar(System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CommentChar(  
    int ch  
)
```

VB

```
Public Sub CommentChar (  
    ch As Integer  
)
```

C++

```
public:  
void CommentChar(  
    int ch  
)
```

F#

```
member CommentChar :  
    ch : int -> unit
```

Parameters

ch

Type: [System.Int32](#)

[Missing <param name="ch"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.CommentChar(System.Int32)"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.EolIsSignificant Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.EolIsSignificant(System.Boolean)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EolIsSignificant (
    bool flag
)
```

VB

```
Public Sub EolIsSignificant (
    flag As Boolean
)
```

C++

```
public:
void EolIsSignificant (
    bool flag
)
```

F#

```
member EolIsSignificant :
    flag : bool -> unit
```

Parameters

flag

Type: [System.Boolean](#)

[Missing <param name="flag"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.EolIsSignificant(System.Boolean)"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<int> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of Integer)
```

C++

```
public:  
virtual IEnumerator<int>^ GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<int>  
override GetEnumerator : unit -> IEnumerator<int>
```

Return Value

Type: [IEnumerator\(Int32\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.NextToken Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.NextToken"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int NextToken()
```

VB

```
Public Function NextToken As Integer
```

C++

```
public:  
int NextToken()
```

F#

```
member NextToken : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.NextToken"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.OrdinaryChar Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.OrdinaryChar(System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OrdinaryChar(  
    int ch  
)
```

VB

```
Public Sub OrdinaryChar (  
    ch As Integer  
)
```

C++

```
public:  
void OrdinaryChar(  
    int ch  
)
```

F#

```
member OrdinaryChar :  
    ch : int -> unit
```

Parameters

ch

Type: [System.Int32](#)

[Missing <param name="ch"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.OrdinaryChar(System.Int32)"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.OrdinaryChars Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.OrdinaryChars(System.Int32,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OrdinaryChars (  
    int low,  
    int hi  
)
```

VB

```
Public Sub OrdinaryChars (  
    low As Integer,  
    hi As Integer  
)
```

C++

```
public:  
void OrdinaryChars(  
    int low,  
    int hi  
)
```

F#

```
member OrdinaryChars :  
    low : int *  
    hi : int -> unit
```

Parameters

low

Type: [System.Int32](#)

[Missing <param name="low"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.OrdinaryChars(System.Int32,System.Int32)"]

hi

Type: [System.Int32](#)

[Missing <param name="hi"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.OrdinaryChars(System.Int32,System.Int32)"]

VelocityDB Class Library

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.ParseNumbers Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.ParseNumbers"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ParseNumbers ()
```

VB

```
Public Sub ParseNumbers
```

C++

```
public:  
void ParseNumbers ()
```

F#

```
member ParseNumbers : unit -> unit
```

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.PushBack Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.PushBack"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void PushBack ()
```

VB

```
Public Sub PushBack
```

C++

```
public:  
void PushBack ()
```

F#

```
member PushBack : unit -> unit
```

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.QuoteChar Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.QuoteChar(System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void QuoteChar(  
    int ch  
)
```

VB

```
Public Sub QuoteChar (  
    ch As Integer  
)
```

C++

```
public:  
void QuoteChar(  
    int ch  
)
```

F#

```
member QuoteChar :  
    ch : int -> unit
```

Parameters

ch

Type: [System.Int32](#)

[Missing <param name="ch"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.QuoteChar(System.Int32)"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.ResetSyntax Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.ResetSyntax"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ResetSyntax ()
```

VB

```
Public Sub ResetSyntax
```

C++

```
public:  
void ResetSyntax ()
```

F#

```
member ResetSyntax : unit -> unit
```

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.ToString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.ToString"]

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.WhitespaceChars Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WhitespaceChars(System.Int32,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void WhitespaceChars (  
    int low,  
    int hi  
)
```

VB

```
Public Sub WhitespaceChars (  
    low As Integer,  
    hi As Integer  
)
```

C++

```
public:  
void WhitespaceChars (  
    int low,  
    int hi  
)
```

F#

```
member WhitespaceChars :  
    low : int *  
    hi : int -> unit
```

Parameters

low

Type: [System.Int32](#)

[Missing <param name="low"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WhitespaceChars(System.Int32,System.Int32)"]

hi

Type: [System.Int32](#)

[Missing <param name="hi"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WhitespaceChars(System.Int32,System.Int32)"]

VelocityDB Class Library

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.WordChars Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WordChars(System.Int32,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void WordChars (  
    int low,  
    int hi  
)
```

VB

```
Public Sub WordChars (  
    low As Integer,  
    hi As Integer  
)
```

C++

```
public:  
void WordChars (  
    int low,  
    int hi  
)
```

F#

```
member WordChars :  
    low : int *  
    hi : int -> unit
```

Parameters

low

Type: [System.Int32](#)

[Missing <param name="low"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WordChars(System.Int32,System.Int32)"]

hi

Type: [System.Int32](#)

[Missing <param name="hi"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.WordChars(System.Int32,System.Int32)"]

VelocityDB Class Library

See Also






[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.StreamTokenizer Fields

The [StreamTokenizer](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | TtEof | |
|  | TtEol | |
|  | TtNumber | |
|  | TtWord | |
|  | Ttype | |

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.TtEof Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.TtEof"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const int TtEof = -1
```

VB

```
Public Const TtEof As Integer = -1
```

C++

```
public:  
literal int TtEof = -1
```

F#

```
static val mutable TtEof: int
```

Field Value

Type: [Int32](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.TtEol Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.TtEol"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const int TtEol = 10
```

VB

```
Public Const TtEol As Integer = 10
```

C++

```
public:  
literal int TtEol = 10
```

F#

```
static val mutable TtEol: int
```

Field Value

Type: [Int32](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.TtNumber Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.TtNumber"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const int TtNumber = -2
```

VB

```
Public Const TtNumber As Integer = -2
```

C++

```
public:  
literal int TtNumber = -2
```

F#

```
static val mutable TtNumber: int
```

Field Value

Type: [Int32](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.TtWord Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.TtWord"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const int TtWord = -3
```

VB

```
Public Const TtWord As Integer = -3
```

C++

```
public:  
literal int TtWord = -3
```

F#

```
static val mutable TtWord: int
```

Field Value

Type: [Int32](#)

See Also

[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

StreamTokenizer.Ttype Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.StreamTokenizer.Ttype"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public int Ttype
```

VB

```
Public Ttype As Integer
```

C++

```
public:  
int Ttype
```

F#

```
val mutable Ttype: int
```

Field Value

Type: [Int32](#)

See Also




[StreamTokenizer Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.IO.GML"]

Classes

| Class | Description |
|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GmlReader | <p>A reader for the Graph Modelling Language (GML).</p> <p>(http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz)</p> <p>It's not clear that all node have to have id's or that they have to be integers - we assume that this is the case. We also assume that only one graph can be defined in a file.</p> |
|  GmlTokens | <p>A collection of tokens used for GML related data.</p> <p>Tokens defined from GML Tags (http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz)</p> |
|  GmlWriter | <p>GMLWriter writes a Graph to a GML OutputStream.</p> <p>GML definition taken from (http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz)</p> |

GmlReader Class

A reader for the Graph Modelling Language (GML).

(<http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz>)

It's not clear that all nodes have to have IDs or that they have to be integers - we assume that this is the case. We also assume that only one graph can be defined in a file.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GmlReader
```

VB

```
Public Class GmlReader
```

C++



```
public ref class GmlReader
```

F#




```
type GmlReader = class end
```

The **GmlReader** type exposes the following members.













Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------|
|  | GmlReader(IGraph) | Create a new GML reader (Uses default edge label DEFAULT_LABEL) |
|  | GmlReader(IGraph, String) | Create a new GML reader |

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------|
|  | EdgeIdKey | gml property to use as id for edges |
|  | EdgeLabelKey | gml property to assign edge labels to |
|  | VertexIdKey | gml property to use as id for vertices |

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(Stream, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|   | InputGraph(IGraph, Stream) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, String) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String, String) | |
|   | InputGraph(IGraph, String, Int32, String, String, String, String) | Load the GML file into the Graph. |

Fields



| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | DefaultLabel | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|------------------------------------------------------------------------|
|  | GmlReader(IGraph) | Create a new GML reader (Uses default edge label DEFAULT_LABEL) |
|  | GmlReader(IGraph, String) | Create a new GML reader |

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader Constructor (IGraph)

Create a new GML reader

(Uses default edge label DEFAULT_LABEL)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GmlReader(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GmlReader(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GmlReader
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to load data into

See Also

[GmlReader Class](#)

[GmlReader Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader Constructor (IGraph, String)

Create a new GML reader

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GmlReader(  
    IGraph graph,  
    string defaultEdgeLabel  
)
```

VB

```
Public Sub New (  
    graph As IGraph,  
    defaultEdgeLabel As String  
)
```

C++

```
public:  
GmlReader(  
    IGraph^ graph,  
    String^ defaultEdgeLabel  
)
```

F#

```
new :  
    graph : IGraph *  
    defaultEdgeLabel : string -> GmlReader
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to load data into

defaultEdgeLabel

Type: [System.String](#)

the default edge label to be used if the GML edge does not define a label

See Also

[GmlReader Class](#)




[GmlReader Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.GmlReader Properties

The [GmlReader](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------|
|  | EdgeIdKey | gml property to use as id for edges |
|  | EdgeLabelKey | gml property to assign edge labels to |
|  | VertexIdKey | gml property to use as id for vertices |

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.EdgeIdKey Property

gml property to use as id for edges

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeIdKey { get; set; }
```

VB

```
Public Property EdgeIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeIdKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.EdgeLabelKey Property

gml property to assign edge labels to

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeLabelKey { get; set; }
```

VB

```
Public Property EdgeLabelKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeLabelKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeLabelKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.VertexIdKey Property

gml property to use as id for vertices

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string VertexIdKey { get; set; }
```

VB

```
Public Property VertexIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ VertexIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member VertexIdKey : string with get, set
```

Property Value

Type: [String](#)









See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.GmlReader Methods

Methods







| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(Stream, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|   | InputGraph(IGraph, Stream) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, String) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String, String) | |
|   | InputGraph(IGraph, String, Int32, String, String, String, String) | Load the GML file into the Graph. |

See Also

[GmlReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method

Overload List

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(Stream, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|  | InputGraph(String, Int32) | Read the GML from from the stream. If the file is malformed incomplete data can be loaded. |
|   | InputGraph(IGraph, Stream) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, String) | Load the GML file into the Graph. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String, String) | |
|   | InputGraph(IGraph, String, Int32, String, String, String, String) | Load the GML file into the Graph. |

See Also

[GmlReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (Stream)

Read the GML from from the stream.

If the file is malformed incomplete data can be loaded.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    Stream inputStream  
)
```

VB

```
Public Sub InputGraph (  
    inputStream As Stream  
)
```

C++

```
public:  
void InputGraph(  
    Stream^ inputStream  
)
```

F#

```
member InputGraph :  
    inputStream : Stream -> unit
```

Parameters

inputStream

Type: [System.IO.Stream](#)

[Missing <param name="inputStream"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.IO.Stream)"]

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (String)

Read the GML from from the stream.

If the file is malformed incomplete data can be loaded.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename  
)
```

VB

```
Public Sub InputGraph (  
    filename As String  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename  
)
```

F#

```
member InputGraph :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

[Missing <param name="filename"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.String)"]

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (Stream, Int32)

Read the GML from from the stream.

If the file is malformed incomplete data can be loaded.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    Stream inputStream,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    inputStream As Stream,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    Stream^ inputStream,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    inputStream : Stream *  
    bufferSize : int -> unit
```

Parameters

inputStream

Type: [System.IO.Stream](#)

[Missing <param name="inputStream"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.IO.Stream,System.Int32)"]

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.IO.Stream,System.Int32)"]

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (String, Int32)

Read the GML from from the stream.

If the file is malformed incomplete data can be loaded.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    filename As String,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    filename : string *  
    bufferSize : int -> unit
```

Parameters

filename

Type: [System.String](#)

[Missing <param name="filename"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.String,System.Int32)"]

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(System.String,System.Int32)"]

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (IGraph, Stream)

Load the GML file into the Graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph graph,  
    Stream inputStream  
)
```

VB

```
Public Shared Sub InputGraph (  
    graph As IGraph,  
    inputStream As Stream  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ graph,  
    Stream^ inputStream  
)
```

F#

```
static member InputGraph :  
    graph : IGraph *  
    inputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

to receive the data

inputStream

Type: [System.IO.Stream](#)

GML file

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (IGraph, String)

Load the GML file into the Graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub InputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member InputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

to receive the data

filename

Type: [System.String](#)

GML file

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (IGraph, Stream, Int32, String, String, String, String)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream inputStream,  
    int bufferSize,  
    string defaultEdgeLabel,  
    string vertexIdKey,  
    string edgeIdKey,  
    string edgeLabelKey  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    inputStream As Stream,  
    bufferSize As Integer,  
    defaultEdgeLabel As String,  
    vertexIdKey As String,  
    edgeIdKey As String,  
    edgeLabelKey As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ inputStream,  
    int bufferSize,  
    String^ defaultEdgeLabel,  
    String^ vertexIdKey,  
    String^ edgeIdKey,  
    String^ edgeLabelKey  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *
```

```
inputStream : Stream *  
bufferSize : int *  
defaultEdgeLabel : string *  
vertexIdKey : string *  
edgeIdKey : string *  
edgeLabelKey : string -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="inputGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

inputStream

Type: [System.IO.Stream](#)

[Missing <param name="inputStream"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

defaultEdgeLabel

Type: [System.String](#)

[Missing <param name="defaultEdgeLabel"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

vertexIdKey

Type: [System.String](#)

[Missing <param name="vertexIdKey"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

edgeIdKey

Type: [System.String](#)

[Missing <param name="edgeIdKey"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac

.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

edgeLabelKey

Type: [System.String](#)

[Missing <param name="edgeLabelKey"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.String,System.String,System.String,System.String)"]

See Also

[GmlReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.InputGraph Method (IGraph, String, Int32, String, String, String, String)

Load the GML file into the Graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename,  
    int bufferSize,  
    string defaultEdgeLabel,  
    string vertexIdKey,  
    string edgeIdKey,  
    string edgeLabelKey  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String,  
    bufferSize As Integer,  
    defaultEdgeLabel As String,  
    vertexIdKey As String,  
    edgeIdKey As String,  
    edgeLabelKey As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename,  
    int bufferSize,  
    String^ defaultEdgeLabel,  
    String^ vertexIdKey,  
    String^ edgeIdKey,  
    String^ edgeLabelKey  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string *  
    bufferSize : int *  
    defaultEdgeLabel : string *  
    vertexIdKey : string *
```

```
edgeIdKey : string *  
edgeLabelKey : string -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

to receive the data

filename

Type: [System.String](#)

GML file

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32,System.String,System.String,System.String,System.String)"]

defaultEdgeLabel

Type: [System.String](#)

default edge label to be used if not defined in the data

vertexIdKey

Type: [System.String](#)

if the id of a vertex is a <data/> property, fetch it from the data property.

edgeIdKey

Type: [System.String](#)

if the id of an edge is a <data/> property, fetch it from the data property.

edgeLabelKey

Type: [System.String](#)

if the label of an edge is a <data/> property, fetch it from the data property.

See Also

[GmlReader Class](#)


[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.GmlReader Fields

The [GmlReader](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | DefaultLabel | |

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlReader.DefaultLabel Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlReader.DefaultLabel"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string DefaultLabel = "undefined"
```

VB

```
Public Const DefaultLabel As String = "undefined"
```

C++

```
public:  
literal String^ DefaultLabel = "undefined"
```

F#

```
static val mutable DefaultLabel: string
```

Field Value

Type: [String](#)

See Also

[GmlReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens Class

A collection of tokens used for GML related data.

Tokens defined from GML Tags (<http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz>)

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GmlTokens
```

VB

```
Public NotInheritable Class GmlTokens
```

C++















```
public ref class GmlTokens abstract sealed
```

F#





```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GmlTokens = class end
```

The **GmlTokens** type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-----------------------------------------------------------------------|
|   | BlueprintsId | Special token used to store Blueprint ids as they may not be integers |
|   | Comment | |
|   | CommentChar | |
|   | Creator | |
|   | Directed | |
|   | Edge | |
|   | Gml | |

VelocityDB Class Library

| | |
|-----------------------------------------------------------------------------------------------------------------|--|
|  Graph | |
|  Graphics | |
|  Id | |
|  Label | |
|  LabelGraphics | |
|  Name | |
|  Node | |
|  Source | |
|  Target | |
|  Version | |


















See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.GmlTokens Fields

The [GmlTokens](#) type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-----------------------------------------------------------------------|
|  | BlueprintsId | Special token used to store Blueprint ids as they may not be integers |
|  | Comment | |
|  | CommentChar | |
|  | Creator | |
|  | Directed | |
|  | Edge | |
|  | Gml | |
|  | Graph | |
|  | Graphics | |
|  | Id | |
|  | Label | |
|  | LabelGraphics | |
|  | Name | |
|  | Node | |
|  | Source | |
|  | Target | |
|  | Version | |

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.BlueprintsId Field

Special token used to store Blueprint ids as they may not be integers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string BlueprintsId = "blueprintsId"
```

VB

```
Public Const BlueprintsId As String = "blueprintsId"
```

C++

```
public:  
literal String^ BlueprintsId = "blueprintsId"
```

F#

```
static val mutable BlueprintsId: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Comment Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Comment"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Comment = "comment"
```

VB

```
Public Const Comment As String = "comment"
```

C++

```
public:  
literal String^ Comment = "comment"
```

F#

```
static val mutable Comment: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.CommentChar Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.CommentChar"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const char CommentChar = '#'
```

VB

```
Public Const CommentChar As Char = "#"C
```

C++

```
public:  
literal wchar_t CommentChar = '#'
```

F#

```
static val mutable CommentChar: char
```

Field Value

Type: [Char](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Creator Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Creator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Creator = "Creator"
```

VB

```
Public Const Creator As String = "Creator"
```

C++

```
public:  
literal String^ Creator = "Creator"
```

F#

```
static val mutable Creator: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Directed Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Directed"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Directed = "directed"
```

VB

```
Public Const Directed As String = "directed"
```

C++

```
public:  
literal String^ Directed = "directed"
```

F#

```
static val mutable Directed: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Edge Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Edge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Edge = "edge"
```

VB

```
Public Const Edge As String = "edge"
```

C++

```
public:  
literal String^ Edge = "edge"
```

F#

```
static val mutable Edge: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Gml Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Gml"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Gml = "gml"
```

VB

```
Public Const Gml As String = "gml"
```

C++

```
public:  
literal String^ Gml = "gml"
```

F#

```
static val mutable Gml: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Graph Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Graph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Graph = "graph"
```

VB

```
Public Const Graph As String = "graph"
```

C++

```
public:  
literal String^ Graph = "graph"
```

F#

```
static val mutable Graph: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Graphics Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Graphics"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Graphics = "graphics"
```

VB

```
Public Const Graphics As String = "graphics"
```

C++

```
public:  
literal String^ Graphics = "graphics"
```

F#

```
static val mutable Graphics: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Id Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Id = "id"
```

VB

```
Public Const Id As String = "id"
```

C++

```
public:  
literal String^ Id = "id"
```

F#

```
static val mutable Id: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Label Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Label = "label"
```

VB

```
Public Const Label As String = "label"
```

C++

```
public:  
literal String^ Label = "label"
```

F#

```
static val mutable Label: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.LabelGraphics Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.LabelGraphics"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string LabelGraphics = "LabelGraphics"
```

VB

```
Public Const LabelGraphics As String = "LabelGraphics"
```

C++

```
public:  
literal String^ LabelGraphics = "LabelGraphics"
```

F#

```
static val mutable LabelGraphics: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Name Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Name = "name"
```

VB

```
Public Const Name As String = "name"
```

C++

```
public:  
literal String^ Name = "name"
```

F#

```
static val mutable Name: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Node Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Node"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Node = "node"
```

VB

```
Public Const Node As String = "node"
```

C++

```
public:  
literal String^ Node = "node"
```

F#

```
static val mutable Node: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Source Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Source"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Source = "source"
```

VB

```
Public Const Source As String = "source"
```

C++

```
public:  
literal String^ Source = "source"
```

F#

```
static val mutable Source: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Target Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Target"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Target = "target"
```

VB

```
Public Const Target As String = "target"
```

C++

```
public:  
literal String^ Target = "target"
```

F#

```
static val mutable Target: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlTokens.Version Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlTokens.Version"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Version = "Version"
```

VB

```
Public Const Version As String = "Version"
```

C++

```
public:  
literal String^ Version = "Version"
```

F#

```
static val mutable Version: string
```

Field Value

Type: [String](#)

See Also

[GmlTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter Class

GmlWriter writes a Graph to a GML OutputStream.

GML definition taken from (<http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz>)

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GML.GmlWriter

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GmlWriter
```

VB

```
Public Class GmlWriter
```

C++


```
public ref class GmlWriter
```

F#



```
type GmlWriter = class end
```




The **GmlWriter** type exposes the following members.

Constructors





| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------|
|  | GmlWriter | Initializes a new instance of the GmlWriter class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | EdgeIdKey | gml property to use for the blueprints edges id, defaults to GMLTokens.BLUEPRINTS_ID |
|  | Normalize | whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs. |

| | |
|---------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Strict | when set to true, property keys in the Graph that do not meet the exact guidelines of the GML specification are ignored. By default this value is false. |
|  UseId | whether to use the blueprints id directly or substitute with a generated integer. To use this option the blueprints ids must all be Integers or string representations of integers |
|  VertexIdKey | gml property to use for the blueprints vertex id, defaults to GMLTokens.BLUEPRINTS_ID |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------|
|  | OutputGraph(Stream) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(String) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(IGraph, String) | Write the data in a Graph to a GML OutputStream. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter Constructor

Initializes a new instance of the [GmlWriter](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GmlWriter(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GmlWriter(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GmlWriter
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

he Graph to pull the data from

See Also






[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.GmlWriter Properties

The [GmlWriter](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | EdgeIdKey | gml property to use for the blueprints edges id, defaults to GMLTokens.BLUEPRINTS_ID |
|  | Normalize | whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs. |
|  | Strict | when set to true, property keys in the Graph that do not meet the exact guidelines of the GML specification are ignored. By default this value is false. |
|  | UseId | whether to use the blueprints id directly or substitute with a generated integer. To use this option the blueprints ids must all be Integers or string representations of integers |
|  | VertexIdKey | gml property to use for the blueprints vertex id, defaults to GMLTokens.BLUEPRINTS_ID |

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.EdgeIdKey Property

gml property to use for the blueprints edges id, defaults to GMLTokens.BLUEPRINTS_ID

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeIdKey { get; set; }
```

VB

```
Public Property EdgeIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeIdKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.Normalize Property

whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Normalize { get; set; }
```

VB

```
Public Property Normalize As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool Normalize {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member Normalize : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.Strict Property

when set to true, property keys in the Graph that do not meet the exact guidelines of the GML specification are ignored. By default this value is false.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Strict { get; set; }
```

VB

```
Public Property Strict As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool Strict {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member Strict : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.UseId Property

whether to use the blueprints id directly or substitute with a generated integer. To use this option the blueprints ids must all be Integers or string representations of integers

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool UseId { get; set; }
```

VB

```
Public Property UseId As Boolean  
    Get  
    Set
```

C++

```
public:  
property bool UseId {  
    bool get ();  
    void set (bool value);  
}
```

F#

```
member UseId : bool with get, set
```

Property Value

Type: [Boolean](#)

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.VertexIdKey Property

gml property to use for the blueprints vertex id, defaults to GMLTokens.BLUEPRINTS_ID

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string VertexIdKey { get; set; }
```

VB

```
Public Property VertexIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ VertexIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member VertexIdKey : string with get, set
```

Property Value

Type: [String](#)





See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.GmlWriter Methods

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------|
|  | OutputGraph(Stream) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(String) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(IGraph, String) | Write the data in a Graph to a GML OutputStream. |







See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.OutputGraph Method

Overload List

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------|
|  | OutputGraph(Stream) | Write the data in a Graph to a GML OutputStream. |
|  | OutputGraph(String) | Write the data in a Graph to a GML OutputStream. |
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GML OutputStream. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a GML OutputStream. |

See Also

[GmlWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.OutputGraph Method (Stream)

Write the data in a Graph to a GML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph(  
    Stream gMlOutputStream  
)
```

VB

```
Public Sub OutputGraph (  
    gMlOutputStream As Stream  
)
```

C++

```
public:  
void OutputGraph(  
    Stream^ gMlOutputStream  
)
```

F#

```
member OutputGraph :  
    gMlOutputStream : Stream -> unit
```

Parameters

gMlOutputStream

Type: [System.IO.Stream](#)

the GML OutputStream to write the Graph data to

See Also

[GmlWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.OutputGraph Method (String)

Write the data in a Graph to a GML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph(  
    string filename  
)
```

VB

```
Public Sub OutputGraph (  
    filename As String  
)
```

C++

```
public:  
void OutputGraph(  
    String^ filename  
)
```

F#

```
member OutputGraph :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

the GML file to write the Graph data to

See Also

[GmlWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.OutputGraph Method (IGraph, Stream)

Write the data in a Graph to a GML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    Stream graphMlOutputStream  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    graphMlOutputStream As Stream  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    Stream^ graphMlOutputStream  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    graphMlOutputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

graphMlOutputStream

Type: [System.IO.Stream](#)

the GML OutputStream to write the Graph data to

See Also

[GmlWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

GmlWriter.OutputGraph Method (IGraph, String)

Write the data in a Graph to a GML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

filename

Type: [System.String](#)

the GML file to write the Graph data to

See Also

[GmlWriter Class](#)




[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GML Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------------------------------------------------------------------|
|  | GraphJsonReader | GraphJsonReader reads the data from a GraphJson JSON stream to a graph. |
|  | GraphJsonSettings | |
|  | GraphJsonWriter | GraphJsonWriter writes a Graph to a GraphJson OutputStream. |

GraphJsonReader Class

GraphJsonReader reads the data from a GraphJson JSON stream to a graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonReader

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphJsonReader
```

VB

```
Public NotInheritable Class GraphJsonReader
```




C++

```
public ref class GraphJsonReader abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphJsonReader = class end
```

Methods







| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|  | InputGraph(IGraph, Stream, Int32) | Input the GraphJson stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(IGraph, String, Int32) | |
|  | InputGraph(IGraph, Stream, Int32, GraphJsonSettings) | Input the GraphJson stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonReader.GraphJsonReader Methods

Methods

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|   | InputGraph(IGraph, Stream, Int32) | Input the GraphJson stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String, Int32) | |
|   | InputGraph(IGraph, Stream, Int32, GraphJsonSettings) | Input the GraphJson stream data into the graph. More control over how data is streamed is provided by this method. |







See Also

[GraphJsonReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonReader.InputGraph Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|   | InputGraph(IGraph, Stream, Int32) | Input the GraphJson stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String, Int32) | |
|   | InputGraph(IGraph, Stream, Int32, GraphJsonSettings) | Input the GraphJson stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[GraphJsonReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonReader.InputGraph Method (IGraph, Stream, Int32)

Input the GraphJson stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph graph,  
    Stream jsonInputStream,  
    int bufferSize = 1000  
)
```

VB

```
Public Shared Sub InputGraph (  
    graph As IGraph,  
    jsonInputStream As Stream,  
    Optional bufferSize As Integer = 1000  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ graph,  
    Stream^ jsonInputStream,  
    int bufferSize = 1000  
)
```

F#

```
static member InputGraph :  
    graph : IGraph *  
    jsonInputStream : Stream *  
    ?bufferSize : int  
(* Defaults:  
    let _bufferSize = defaultArg bufferSize 1000  
*)  
-> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphJson data

jsonInputStream

Type: [System.IO.Stream](#)

a Stream of JSON data

VelocityDB Class Library

bufferSize (Optional)

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

See Also

[GraphJsonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonReader.InputGraph Method (IGraph, String, Int32)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename,  
    int bufferSize = 1000  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String,  
    Optional bufferSize As Integer = 1000  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename,  
    int bufferSize = 1000  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string *  
    ?bufferSize : int  
(* Defaults:  
    let _bufferSize = defaultArg bufferSize 1000  
)  
-> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="inputGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

filename

Type: [System.String](#)

[Missing <param name="filename"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

bufferSize (Optional)

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

See Also

[GraphJsonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonReader.InputGraph Method (IGraph, Stream, Int32, GraphJsonSettings)

Input the GraphJson stream data into the graph. More control over how data is streamed is provided by this method.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream jsonInputStream,  
    int bufferSize,  
    GraphJsonSettings settings  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    jsonInputStream As Stream,  
    bufferSize As Integer,  
    settings As GraphJsonSettings  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ jsonInputStream,  
    int bufferSize,  
    GraphJsonSettings^ settings  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    jsonInputStream : Stream *  
    bufferSize : int *  
    settings : GraphJsonSettings -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphJson data

jsonInputStream

VelocityDB Class Library

Type: [System.IO.Stream](#)

a Stream of GraphJson data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

settings

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings](#)

Contains field names that the reader will use to parse the graph

See Also

[GraphJsonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class GraphJsonSettings
```


```
VB
Public Class GraphJsonSettings
```

```
C++
public ref class GraphJsonSettings
```







```
F#
type GraphJsonSettings = class end
```

The **GraphJsonSettings** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | GraphJsonSettings | Initializes a new instance of the GraphJsonSettings class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|-------------|
|  | Default | |
|  | EdgeCaptionProp | |
|  | IdProp | |
|  | NodeCaptionProp | |
|  | SourceProp | |
|  | TargetProp | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings Constructor

Initializes a new instance of the [GraphJsonSettings](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphJsonSettings ()
```

VB

```
Public Sub New
```

C++

```
public:  
GraphJsonSettings ()
```

F#

```
new : unit -> GraphJsonSettings
```

See Also







[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.GraphJsonSettings Properties

The [GraphJsonSettings](#) type exposes the following members.

Properties

| | Name | Description |
|--------------------------------------------------------------------------------------------|---------------------------------|-------------|
|  S | Default | |
|  | EdgeCaptionProp | |
|  | IdProp | |
|  | NodeCaptionProp | |
|  | SourceProp | |
|  | TargetProp | |

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.Default Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.Default"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static GraphJsonSettings Default { get; }
```

VB

```
Public Shared ReadOnly Property Default As GraphJsonSettings  
    Get
```

C++

```
public:  
static property GraphJsonSettings^ Default {  
    GraphJsonSettings^ get ();  
}
```

F#

```
static member Default : GraphJsonSettings with get
```

Property Value

Type: [GraphJsonSettings](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.EdgeCaptionProp Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.EdgeCaptionProp"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeCaptionProp { get; set; }
```

VB

```
Public Property EdgeCaptionProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeCaptionProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeCaptionProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.IdProp Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.IdProp"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string IdProp { get; set; }
```

VB

```
Public Property IdProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ IdProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member IdProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.NodeCaptionProp Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.NodeCaptionProp"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string NodeCaptionProp { get; set; }
```

VB

```
Public Property NodeCaptionProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ NodeCaptionProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member NodeCaptionProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.SourceProp Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.SourceProp"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string SourceProp { get; set; }
```

VB

```
Public Property SourceProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ SourceProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member SourceProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonSettings.TargetProp Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings.TargetProp"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string TargetProp { get; set; }
```

VB

```
Public Property TargetProp As String  
    Get  
    Set
```

C++

```
public:  
property String^ TargetProp {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member TargetProp : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphJsonSettings Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter Class

GraphJsonWriter writes a Graph to a GraphJson OutputStream.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonWriter

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphJsonWriter
```

VB

```
Public NotInheritable Class GraphJsonWriter
```









C++

```
public ref class GraphJsonWriter abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphJsonWriter = class end
```

Methods





| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------|
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, String, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.GraphJsonWriter Methods

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------|
|  | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphJson OutputStream. |
|  | OutputGraph(IGraph, String) | Write the data in a Graph to a GraphJson OutputStream. |
|  | OutputGraph(IGraph, Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |
|  | OutputGraph(IGraph, String, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |









See Also

[GraphJsonWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.OutputGraph Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------|
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, Stream, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |
|   | OutputGraph(IGraph, String, GraphJsonSettings) | Write the data in a Graph to a GraphJson OutputStream. |

See Also

[GraphJsonWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.OutputGraph Method (IGraph, Stream)

Write the data in a Graph to a GraphJson OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    Stream jsonOutputStream  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    jsonOutputStream As Stream  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    Stream^ jsonOutputStream  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    jsonOutputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize

jsonOutputStream

Type: [System.IO.Stream](#)

the OutputStream to write to

See Also

[GraphJsonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.OutputGraph Method (IGraph, String)

Write the data in a Graph to a GraphJson OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize

filename

Type: [System.String](#)

the JSON file to write the Graph data to

See Also

[GraphJsonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.OutputGraph Method (IGraph, Stream, GraphJsonSettings)

Write the data in a Graph to a GraphJson OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    Stream jsonOutputStream,
    GraphJsonSettings settings
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    jsonOutputStream As Stream,
    settings As GraphJsonSettings
)
```

C++

```
public:
static void OutputGraph (
    IGraph^ graph,
    Stream^ jsonOutputStream,
    GraphJsonSettings^ settings
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    jsonOutputStream : Stream *
    settings : GraphJsonSettings -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize

jsonOutputStream

Type: [System.IO.Stream](#)

the OutputStream to write to

settings

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings](#)

Contains field names that the writer will use to map to BluePrints

See Also

[GraphJsonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

GraphJsonWriter.OutputGraph Method (IGraph, String, GraphJsonSettings)

Write the data in a Graph to a GraphJson OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename,  
    GraphJsonSettings settings  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String,  
    settings As GraphJsonSettings  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename,  
    GraphJsonSettings^ settings  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string *  
    settings : GraphJsonSettings -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize

filename

Type: [System.String](#)

the JSON file to write the Graph data to

settings

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson.GraphJsonSettings](#)

Contains field names that the writer will use to map to BluePrints

See Also

[GraphJsonWriter Class](#)





[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphJson Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------------------|
|  | GraphMigrator | GraphMigrator takes the data in one graph and pipes it to another graph. |
|  | GraphMLReader | GraphMLReader writes the data from a GraphML stream to a graph. |
|  | GraphMLTokens | A collection of tokens used for GraphML related data. |
|  | GraphMLWriter | GraphMLWriter writes a Graph to a GraphML OutputStream. |

GraphMigrator Class

GraphMigrator takes the data in one graph and pipes it to another graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMigrator

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphMigrator
```

VB

```
Public NotInheritable Class GraphMigrator
```

C++


```
public ref class GraphMigrator abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphMigrator = class end
```

The **GraphMigrator** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|------------------------------------------------|
|  | MigrateGraph | Pipe the data from one graph to another graph. |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMigrator.GraphMigrator Methods

The [GraphMigrator](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|------------------------------------------------|
|  | MigrateGraph | Pipe the data from one graph to another graph. |

See Also

[GraphMigrator Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMigrator.MigrateGraph Method

Pipe the data from one graph to another graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void MigrateGraph(  
    IGraph fromGraph,  
    IGraph toGraph  
)
```

VB

```
Public Shared Sub MigrateGraph (  
    fromGraph As IGraph,  
    toGraph As IGraph  
)
```

C++

```
public:  
static void MigrateGraph(  
    IGraph^ fromGraph,  
    IGraph^ toGraph  
)
```

F#

```
static member MigrateGraph :  
    fromGraph : IGraph *  
    toGraph : IGraph -> unit
```

Parameters

fromGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to take data from

toGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to take data to

See Also

[GraphMigrator Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader Class

GraphMLReader writes the data from a GraphML stream to a graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLReader

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphMLReader
```

VB

```
Public Class GraphMLReader
```

C++


```
public ref class GraphMLReader
```

F#




```
type GraphMLReader = class end
```

The **GraphMLReader** type exposes the following members.


Constructors












| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | GraphMLReader | Initializes a new instance of the GraphMLReader class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------|
|  | EdgeIdKey | if the id of an edge is a <data/> property, fetch it from the data property. |
|  | EdgeLabelKey | if the label of an edge is a <data/> property, fetch it from the data property. |
|  | VertexIdKey | if the id of a vertex is a <data/> property, fetch it from the data property. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | InputGraph(String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader Constructor

Initializes a new instance of the [GraphMLReader](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphMLReader (  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GraphMLReader (  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GraphMLReader
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphML data

See Also




[GraphMLReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.GraphMLReader Properties

The [GraphMLReader](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------|
|  | EdgeIdKey | if the id of an edge is a <data/> property, fetch it from the data property. |
|  | EdgeLabelKey | if the label of an edge is a <data/> property, fetch it from the data property. |
|  | VertexIdKey | if the id of a vertex is a <data/> property, fetch it from the data property. |

See Also

[GraphMLReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.EdgeIdKey Property

if the id of an edge is a <data/> property, fetch it from the data property.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeIdKey { get; set; }
```

VB

```
Public Property EdgeIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeIdKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphMLReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.EdgeLabelKey Property

if the label of an edge is a <data/> property, fetch it from the data property.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string EdgeLabelKey { get; set; }
```

VB

```
Public Property EdgeLabelKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ EdgeLabelKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member EdgeLabelKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[GraphMLReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.VertexIdKey Property

if the id of a vertex is a <data/> property, fetch it from the data property.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string VertexIdKey { get; set; }
```

VB

```
Public Property VertexIdKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ VertexIdKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member VertexIdKey : string with get, set
```

Property Value

Type: [String](#)













See Also

[GraphMLReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.GraphMLReader Methods

Methods













| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[GraphMLReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method

Overload List

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the GraphML stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, String, String, String) | Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[GraphMLReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (Stream)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public void InputGraph(  
    Stream graphMLInputStream  
)
```

```
VB  
Public Sub InputGraph (  
    graphMLInputStream As Stream  
)
```

```
C++  
public:  
void InputGraph(  
    Stream^ graphMLInputStream  
)
```

```
F#  
member InputGraph :  
    graphMLInputStream : Stream -> unit
```

Parameters

graphMLInputStream

Type: [System.IO.Stream](#)

a Stream of GraphML data

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (String)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename  
)
```

VB

```
Public Sub InputGraph (  
    filename As String  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename  
)
```

F#

```
member InputGraph :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file containing GraphML data

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (Stream, Int32)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    Stream graphMLInputStream,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    graphMLInputStream As Stream,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    Stream^ graphMLInputStream,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    graphMLInputStream : Stream *  
    bufferSize : int -> unit
```

Parameters

graphMLInputStream

Type: [System.IO.Stream](#)

a Stream of GraphML data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (String, Int32)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    filename As String,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    filename : string *  
    bufferSize : int -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file containing GraphML data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (IGraph, Stream)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream graphMLInputStream  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    graphMLInputStream As Stream  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ graphMLInputStream  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    graphMLInputStream : Stream -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphML data

graphMLInputStream

Type: [System.IO.Stream](#)

a Stream of GraphML data

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (IGraph, String)

Input the GraphML stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphML data

filename

Type: [System.String](#)

name of a file containing GraphML data

See Also

[GraphMLReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (IGraph, Stream, Int32, String, String, String)

Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream graphMLInputStream,  
    int bufferSize,  
    string vertexIdKey,  
    string edgeIdKey,  
    string edgeLabelKey  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    graphMLInputStream As Stream,  
    bufferSize As Integer,  
    vertexIdKey As String,  
    edgeIdKey As String,  
    edgeLabelKey As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ graphMLInputStream,  
    int bufferSize,  
    String^ vertexIdKey,  
    String^ edgeIdKey,  
    String^ edgeLabelKey  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    graphMLInputStream : Stream *  
    bufferSize : int *  
    vertexIdKey : string *  
    edgeIdKey : string *  
    edgeLabelKey : string -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphML data

graphMLInputStream

Type: [System.IO.Stream](#)

a Stream of GraphML data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

vertexIdKey

Type: [System.String](#)

if the id of a vertex is a <data/> property, fetch it from the data property.

edgeIdKey

Type: [System.String](#)

if the id of an edge is a <data/> property, fetch it from the data property.

edgeLabelKey

Type: [System.String](#)

if the label of an edge is a <data/> property, fetch it from the data property.

See Also

[GraphMIReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLReader.InputGraph Method (IGraph, String, Int32, String, String, String)

Input the GraphML stream data into the graph. More control over how data is streamed is provided by this method.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename,  
    int bufferSize,  
    string vertexIdKey,  
    string edgeIdKey,  
    string edgeLabelKey  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String,  
    bufferSize As Integer,  
    vertexIdKey As String,  
    edgeIdKey As String,  
    edgeLabelKey As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename,  
    int bufferSize,  
    String^ vertexIdKey,  
    String^ edgeIdKey,  
    String^ edgeLabelKey  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string *  
    bufferSize : int *  
    vertexIdKey : string *  
    edgeIdKey : string *  
    edgeLabelKey : string -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the GraphML data

filename

Type: [System.String](#)

name of a file containing GraphML data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

vertexIdKey

Type: [System.String](#)

if the id of a vertex is a <data/> property, fetch it from the data property.

edgeIdKey

Type: [System.String](#)

if the id of an edge is a <data/> property, fetch it from the data property.

edgeLabelKey

Type: [System.String](#)

if the label of an edge is a <data/> property, fetch it from the data property.

See Also

[GraphMIReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens Class

A collection of tokens used for GraphML related data.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphMlTokens
```

VB

```
Public NotInheritable Class GraphMlTokens
```

C++












```
public ref class GraphMlTokens abstract sealed
```

F#


```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphMlTokens = class end
```

The **GraphMITokens** type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------|-------------|
|  | AttrName | |
|  | AttrType | |
|  | Boolean | |
|  | Data | |
|  | Default | |
|  | DefaultGraphmlSchemaLocation | |
|  | Directed | |
|  | Double | |
|  | Edge | |
|  | Edgedefault | |
|  | Float | |

VelocityDB Class Library

| | | |
|---------------------------------------------------------------------------------------------|--------------------------------------------|--|
|  S | For | |
|  S | G | |
|  S | Graph | |
|  S | Graphml | |
|  S | GraphmlXmlns | |
|  S | Id | |
|  S | Int | |
|  S | Key | |
|  S | Label | |
|  S | Long | |
|  S | Node | |
|  S | Source | |
|  S | String | |
|  S | Target | |
|  S | Xmlns | |
|  S | XmlSchemaLocationAttribute | |
|  S | XmlSchemaNamespaceTag | |


















See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.GraphMITokens Fields

The [GraphMITokens](#) type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|-------------|
|   | AttrName | |
|   | AttrType | |
|   | Boolean | |
|   | Data | |
|   | Default | |
|   | DefaultGraphmlSchemaLocation | |
|   | Directed | |
|   | Double | |
|   | Edge | |
|   | Edgedefault | |
|   | Float | |
|   | For | |
|   | G | |
|   | Graph | |
|   | Graphml | |
|   | GraphmlXmlns | |
|   | Id | |
|   | Int | |
|   | Key | |
|   | Label | |
|   | Long | |
|   | Node | |
|   | Source | |
|   | String | |
|   | Target | |
|   | Xmlns | |
|   | XmlSchemaLocationAttribute | |
|   | XmlSchemaNamespaceTag | |

See Also

[GraphMITokens Class](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.AttrName Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.AttrName"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string AttrName = "attr.name"
```

VB

```
Public Const AttrName As String = "attr.name"
```

C++

```
public:  
literal String^ AttrName = "attr.name"
```

F#

```
static val mutable AttrName: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.AttrType Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.AttrType"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string AttrType = "attr.type"
```

VB

```
Public Const AttrType As String = "attr.type"
```

C++

```
public:  
literal String^ AttrType = "attr.type"
```

F#

```
static val mutable AttrType: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Boolean Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Boolean"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Boolean = "boolean"
```

VB

```
Public Const Boolean As String = "boolean"
```

C++

```
public:  
literal String^ Boolean = "boolean"
```

F#

```
static val mutable Boolean: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Data Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Data"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Data = "data"
```

VB

```
Public Const Data As String = "data"
```

C++

```
public:  
literal String^ Data = "data"
```

F#

```
static val mutable Data: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Default Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Default"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Default = "_default"
```

VB

```
Public Const Default As String = "_default"
```

C++

```
public:  
literal String^ Default = "_default"
```

F#

```
static val mutable Default: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.DefaultGraphmlSchemaLocation Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.DefaultGraphmlSchemaLocation"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string DefaultGraphmlSchemaLocation =  
"http://graphml.graphdrawing.org/xmlns/1.1/graphml.xsd"
```

VB

```
Public Const DefaultGraphmlSchemaLocation As String =  
"http://graphml.graphdrawing.org/xmlns/1.1/graphml.xsd"
```

C++

```
public:  
literal String^ DefaultGraphmlSchemaLocation =  
"http://graphml.graphdrawing.org/xmlns/1.1/graphml.xsd"
```

F#

```
static val mutable DefaultGraphmlSchemaLocation: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Directed Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Directed"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Directed = "directed"
```

VB

```
Public Const Directed As String = "directed"
```

C++

```
public:  
literal String^ Directed = "directed"
```

F#

```
static val mutable Directed: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Double Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Double"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Double = "double"
```

VB

```
Public Const Double As String = "double"
```

C++

```
public:  
literal String^ Double = "double"
```

F#

```
static val mutable Double: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Edge Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Edge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Edge = "edge"
```

VB

```
Public Const Edge As String = "edge"
```

C++

```
public:  
literal String^ Edge = "edge"
```

F#

```
static val mutable Edge: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Edgesdefault Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Edgesdefault"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Edgesdefault = "edgesdefault"
```

VB

```
Public Const Edgesdefault As String = "edgesdefault"
```

C++

```
public:  
literal String^ Edgesdefault = "edgesdefault"
```

F#

```
static val mutable Edgesdefault: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Float Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Float"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Float = "float"
```

VB

```
Public Const Float As String = "float"
```

C++

```
public:  
literal String^ Float = "float"
```

F#

```
static val mutable Float: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.For Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.For"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string For = "for"
```

VB

```
Public Const For As String = "for"
```

C++

```
public:  
literal String^ For = "for"
```

F#

```
static val mutable For: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.G Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.G"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string G = "G"
```

VB

```
Public Const G As String = "G"
```

C++

```
public:  
literal String^ G = "G"
```

F#

```
static val mutable G: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Graph Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Graph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Graph = "graph"
```

VB

```
Public Const Graph As String = "graph"
```

C++

```
public:  
literal String^ Graph = "graph"
```

F#

```
static val mutable Graph: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Graphml Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Graphml"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Graphml = "graphml"
```

VB

```
Public Const Graphml As String = "graphml"
```

C++

```
public:  
literal String^ Graphml = "graphml"
```

F#

```
static val mutable Graphml: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.GraphmlXmlns Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.GraphmlXmlns"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string GraphmlXmlns = "http://graphml.graphdrawing.org/xmlns"
```

VB

```
Public Const GraphmlXmlns As String = "http://graphml.graphdrawing.org/xmlns"
```

C++

```
public:  
literal String^ GraphmlXmlns = "http://graphml.graphdrawing.org/xmlns"
```

F#

```
static val mutable GraphmlXmlns: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Id Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Id = "id"
```

VB

```
Public Const Id As String = "id"
```

C++

```
public:  
literal String^ Id = "id"
```

F#

```
static val mutable Id: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Int Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Int"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Int = "int"
```

VB

```
Public Const Int As String = "int"
```

C++

```
public:  
literal String^ Int = "int"
```

F#

```
static val mutable Int: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Key Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Key"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Key = "key"
```

VB

```
Public Const Key As String = "key"
```

C++

```
public:  
literal String^ Key = "key"
```

F#

```
static val mutable Key: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Label Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Label = "label"
```

VB

```
Public Const Label As String = "label"
```

C++

```
public:  
literal String^ Label = "label"
```

F#

```
static val mutable Label: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Long Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Long"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Long = "long"
```

VB

```
Public Const Long As String = "long"
```

C++

```
public:  
literal String^ Long = "long"
```

F#

```
static val mutable Long: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Node Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Node"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Node = "node"
```

VB

```
Public Const Node As String = "node"
```

C++

```
public:  
literal String^ Node = "node"
```

F#

```
static val mutable Node: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Source Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Source"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Source = "source"
```

VB

```
Public Const Source As String = "source"
```

C++

```
public:  
literal String^ Source = "source"
```

F#

```
static val mutable Source: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.String Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.String"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string String = "string"
```

VB

```
Public Const String As String = "string"
```

C++

```
public:  
literal String^ String = "string"
```

F#

```
static val mutable String: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Target Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Target"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Target = "target"
```

VB

```
Public Const Target As String = "target"
```

C++

```
public:  
literal String^ Target = "target"
```

F#

```
static val mutable Target: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.Xmlns Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.Xmlns"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Xmlns = "xmlns"
```

VB

```
Public Const Xmlns As String = "xmlns"
```

C++

```
public:  
literal String^ Xmlns = "xmlns"
```

F#

```
static val mutable Xmlns: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.XmlSchemaLocationAttribute Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.XmlSchemaLocationAttribute"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string XmlSchemaLocationAttribute = "schemaLocation"
```

VB

```
Public Const XmlSchemaLocationAttribute As String = "schemaLocation"
```

C++

```
public:  
literal String^ XmlSchemaLocationAttribute = "schemaLocation"
```

F#

```
static val mutable XmlSchemaLocationAttribute: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMITokens.XmlSchemaNamespaceTag Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMITokens.XmlSchemaNamespaceTag"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string XmlSchemaNamespaceTag = "xsi"
```

VB

```
Public Const XmlSchemaNamespaceTag As String = "xsi"
```

C++

```
public:  
literal String^ XmlSchemaNamespaceTag = "xsi"
```

F#

```
static val mutable XmlSchemaNamespaceTag: string
```

Field Value

Type: [String](#)

See Also

[GraphMITokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter Class

GraphMLWriter writes a Graph to a GraphML OutputStream.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLWriter

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphMLWriter
```

VB

```
Public Class GraphMLWriter
```

C++


```
public ref class GraphMLWriter
```

F#









```
type GraphMLWriter = class end
```








The **GraphMLWriter** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | GraphMLWriter | Initializes a new instance of the GraphMLWriter class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------|
|  | OutputGraph(Stream) | Write the data in a Graph to a GraphML OutputStream. |
|  | OutputGraph(String) | Write the data in a Graph to a GraphML file. |
|  | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphML OutputStream. |
|  | | |
|  | OutputGraph(IGraph, String) | Write the data in a Graph to a GraphML file. |
|  | | |
|  | OutputGraph(IGraph, Stream, Dictionary(String, String), Dictionary(String, String)) | Write the data in a Graph to a GraphML OutputStream. |
|  | | |

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | OutputGraph(IGraph, String, Dictionary(String, String), Dictionary(String, String)) | <p>Write the data in a Graph to a GraphML file.</p> |
|  | SetEdgeKeyTypes | <p>a IDictionary<string, string> of the data types of the edge keys</p> |
|  | SetEdgeLabelKey | <p>Set the name of the edge label in the GraphML. When this value is not set the value of the Edge.getLabel() is written as a "label" attribute on the edge element. This does not validate against the GraphML schema. If this value is set then the the value of Edge.getLabel() is written as a data element on the edge and the appropriate key element is added to define it in the GraphML</p> |
|  | SetNormalize | <p>whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs.</p> |
|  | SetVertexKeyTypes | <p>a IDictionary<string, string> of the data types of the vertex keys</p> |
|  | SetXmlSchemaLocation | <p>the location of the GraphML XML Schema instance</p> |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter Constructor

Initializes a new instance of the [GraphMLWriter](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphMLWriter(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GraphMLWriter(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GraphMLWriter
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

See Also












[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.GraphMLWriter Methods

The [GraphMLWriter](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  OutputGraph(Stream) | Write the data in a Graph to a GraphML OutputStream. |
|  OutputGraph(String) | Write the data in a Graph to a GraphML file. |
|  OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphML OutputStream. |
|  OutputGraph(IGraph, String) | Write the data in a Graph to a GraphML file. |
|  OutputGraph(IGraph, Stream, Dictionary(String, String), Dictionary(String, String)) | Write the data in a Graph to a GraphML OutputStream. |
|  OutputGraph(IGraph, String, Dictionary(String, String), Dictionary(String, String)) | Write the data in a Graph to a GraphML file. |
|  SetEdgeKeyTypes | a IDictionary<string, string> of the data types of the edge keys |
|  SetEdgeLabelKey | Set the name of the edge label in the GraphML. When this value is not set the value of the Edge.getLabel() is written as a "label" attribute on the edge element. This does not validate against the GraphML schema. If this value is set then the the value of Edge.getLabel() is written as a data element on the edge and the appropriate key element is added to define it in the GraphML |
|  SetNormalize | whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs. |
|  SetVertexKeyTypes | a IDictionary<string, string> of the data types of the vertex keys |
|  SetXmlSchemaLocation | the location of the GraphML XML Schema instance |











See Also

[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------|
|  | OutputGraph(Stream) | Write the data in a Graph to a GraphML OutputStream. |
|  | OutputGraph(String) | Write the data in a Graph to a GraphML file. |
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a GraphML OutputStream. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a GraphML file. |
|   | OutputGraph(IGraph, Stream, Dictionary(String, String), Dictionary(String, String)) | Write the data in a Graph to a GraphML OutputStream. |
|   | OutputGraph(IGraph, String, Dictionary(String, String), Dictionary(String, String)) | Write the data in a Graph to a GraphML file. |

See Also

[GraphMLWriter Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (Stream)

Write the data in a Graph to a GraphML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph(  
    Stream graphMLOutputStream  
)
```

VB

```
Public Sub OutputGraph (  
    graphMLOutputStream As Stream  
)
```

C++

```
public:  
void OutputGraph(  
    Stream^ graphMLOutputStream  
)
```

F#

```
member OutputGraph :  
    graphMLOutputStream : Stream -> unit
```

Parameters

graphMLOutputStream

Type: [System.IO.Stream](#)

the GraphML OutputStream to write the Graph data to

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (String)

Write the data in a Graph to a GraphML file.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph(  
    string filename  
)
```

VB

```
Public Sub OutputGraph (  
    filename As String  
)
```

C++

```
public:  
void OutputGraph(  
    String^ filename  
)
```

F#

```
member OutputGraph :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

the name of the file write the Graph data (as GraphML) to

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (IGraph, Stream)

Write the data in a Graph to a GraphML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    Stream graphMLOutputStream  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    graphMLOutputStream As Stream  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    Stream^ graphMLOutputStream  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    graphMLOutputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

graphMLOutputStream

Type: [System.IO.Stream](#)

the GraphML OutputStream to write the Graph data to

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (IGraph, String)

Write the data in a Graph to a GraphML file.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

filename

Type: [System.String](#)

the name of the file write the Graph data (as GraphML) to

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (IGraph, Stream, Dictionary(String, String), Dictionary(String, String))

Write the data in a Graph to a GraphML OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    Stream graphMLOutputStream,
    Dictionary<string, string> vertexKeyTypes,
    Dictionary<string, string> edgeKeyTypes
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    graphMLOutputStream As Stream,
    vertexKeyTypes As Dictionary(Of String, String),
    edgeKeyTypes As Dictionary(Of String, String)
)
```

C++

```
public:
static void OutputGraph (
    IGraph^ graph,
    Stream^ graphMLOutputStream,
    Dictionary<String^, String^>^ vertexKeyTypes,
    Dictionary<String^, String^>^ edgeKeyTypes
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    graphMLOutputStream : Stream *
    vertexKeyTypes : Dictionary<string, string> *
    edgeKeyTypes : Dictionary<string, string> -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

graphMLOutputStream

Type: [System.IO.Stream](#)

the GraphML OutputStream to write the Graph data to

vertexKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

a IDictionary<string, string> of the data types of the vertex keys

edgeKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

a IDictionary<string, string> of the data types of the edge keys

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.OutputGraph Method (IGraph, String, Dictionary(String, String), Dictionary(String, String))

Write the data in a Graph to a GraphML file.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    string filename,
    Dictionary<string, string> vertexKeyTypes,
    Dictionary<string, string> edgeKeyTypes
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    filename As String,
    vertexKeyTypes As Dictionary(Of String, String),
    edgeKeyTypes As Dictionary(Of String, String)
)
```

C++

```
public:
static void OutputGraph (
    IGraph^ graph,
    String^ filename,
    Dictionary<String^, String^>^ vertexKeyTypes,
    Dictionary<String^, String^>^ edgeKeyTypes
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    filename : string *
    vertexKeyTypes : Dictionary<string, string> *
    edgeKeyTypes : Dictionary<string, string> -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from

filename

Type: [System.String](#)

the name of the file write the Graph data (as GraphML) to

vertexKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

a IDictionary<string, string> of the data types of the vertex keys

edgeKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

a IDictionary<string, string> of the data types of the edge keys

See Also

[GraphMLWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.SetEdgeKeyTypes Method

a IDictionary<string, string> of the data types of the edge keys

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetEdgeKeyTypes (  
    Dictionary<string, string> edgeKeyTypes  
)
```

VB

```
Public Sub SetEdgeKeyTypes (  
    edgeKeyTypes As Dictionary(Of String, String)  
)
```

C++

```
public:  
void SetEdgeKeyTypes (  
    Dictionary<String^, String^>^ edgeKeyTypes  
)
```

F#

```
member SetEdgeKeyTypes :  
    edgeKeyTypes : Dictionary<string, string> -> unit
```

Parameters

edgeKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

[Missing <param name="edgeKeyTypes"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLWriter.SetEdgeKeyTypes(System.Collections.Generic.Dictionary{System.String,System.String})"]

See Also

[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.SetEdgeLabelKey Method

Set the name of the edge label in the GraphML. When this value is not set the value of the `Edge.getLabel()` is written as a "label" attribute on the edge element. This does not validate against the GraphML schema. If this value is set then the the value of `Edge.getLabel()` is written as a data element on the edge and the appropriate key element is added to define it in the GraphML

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetEdgeLabelKey(  
    string edgeLabelKey  
)
```

VB

```
Public Sub SetEdgeLabelKey (  
    edgeLabelKey As String  
)
```

C++

```
public:  
void SetEdgeLabelKey(  
    String^ edgeLabelKey  
)
```

F#

```
member SetEdgeLabelKey :  
    edgeLabelKey : string -> unit
```

Parameters

edgeLabelKey

Type: [System.String](#)

if the label of an edge will be handled by the data property.

See Also

[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.SetNormalize Method

whether to normalize the output. Normalized output is deterministic with respect to the order of elements and properties in the resulting XML document, and is compatible with line diff-based tools such as Git. Note: normalized output is memory-intensive and is not appropriate for very large graphs.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetNormalize (  
    bool normalize  
)
```

VB

```
Public Sub SetNormalize (  
    normalize As Boolean  
)
```

C++

```
public:  
void SetNormalize (  
    bool normalize  
)
```

F#

```
member SetNormalize :  
    normalize : bool -> unit
```

Parameters

normalize

Type: [System.Boolean](#)

[Missing <param name="normalize"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLWriter.SetNormalize(System.Boolean)"]

See Also

[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.SetVertexKeyTypes Method

a IDictionary<string, string> of the data types of the vertex keys

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetVertexKeyTypes (  
    Dictionary<string, string> vertexKeyTypes  
)
```

VB

```
Public Sub SetVertexKeyTypes (  
    vertexKeyTypes As Dictionary(Of String, String)  
)
```

C++

```
public:  
void SetVertexKeyTypes (  
    Dictionary<String^, String^>^ vertexKeyTypes  
)
```

F#

```
member SetVertexKeyTypes :  
    vertexKeyTypes : Dictionary<string, string> -> unit
```

Parameters

vertexKeyTypes

Type: [System.Collections.Generic.Dictionary\(String, String\)](#)

[Missing <param name="vertexKeyTypes"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLWriter.SetVertexKeyTypes(System.Collections.Generic.Dictionary{System.String,System.String})"]

See Also

[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)

GraphMLWriter.SetXmlSchemaLocation Method

the location of the GraphML XML Schema instance

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetXmlSchemaLocation(  
    string xmlSchemaLocation  
)
```

VB

```
Public Sub SetXmlSchemaLocation (  
    xmlSchemaLocation As String  
)
```

C++

```
public:  
void SetXmlSchemaLocation(  
    String^ xmlSchemaLocation  
)
```

F#

```
member SetXmlSchemaLocation :  
    xmlSchemaLocation : string -> unit
```

Parameters

xmlSchemaLocation

Type: [System.String](#)

[Missing <param name="xmlSchemaLocation"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML.GraphMLWriter.SetXmlSchemaLocation(System.String)"]

See Also








[GraphMLWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphML Namespace](#)


VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON"]



Classes

| Class | Description |
|--------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
|  ElementFactoryContract | |
|  ElementPropertyConfig | Configure how the GraphSON utility treats edge and vertex properties. |
|  GraphElementFactory | The standard factory used for most graph element creation. It uses an actual Graph implementation to construct vertices and edges |
|  GraphSonReader | GraphSONReader reads the data from a TinkerPop JSON stream to a graph. |
|  GraphSonTokens | |
|  GraphSonUtility | Helps write individual graph elements to TinkerPop JSON format known as GraphSON. |
|  GraphSonWriter | GraphSONWriter writes a Graph to a TinkerPop JSON OutputStream. |

Interfaces

| Interface | Description |
|---------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IElementFactory | A factory responsible for creating graph elements. Abstracts the way that graph elements are created. In most cases a Graph is responsible for element creation, but there are cases where more control over how vertices and edges are constructed. |

Enumerations

| Enumeration | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
|  ElementPropertyConfig.ElementPropertiesRule | |
|  GraphSonMode | Modes of operation of the GraphSONUtility. |

ElementFactoryContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class ElementFactoryContract
```

VB

```
Public NotInheritable Class ElementFactoryContract
```

C++

```
public ref class ElementFactoryContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type ElementFactoryContract = class end
```

The **ElementFactoryContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------|
|  | ValidateCreateEdge | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementFactoryContract.ElementFactoryContract Methods

The [ElementFactoryContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------|
|  | ValidateCreateEdge | |

See Also

[ElementFactoryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementFactoryContract.ValidateCreateEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract.ValidateCreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateCreateEdge (
    Object id,
    IVertex out_,
    IVertex in_,
    string label
)
```

VB

```
Public Shared Sub ValidateCreateEdge (
    id As Object,
    out_ As IVertex,
    in_ As IVertex,
    label As String
)
```

C++

```
public:
static void ValidateCreateEdge (
    Object^ id,
    IVertex^ out_,
    IVertex^ in_,
    String^ label
)
```

F#

```
static member ValidateCreateEdge :
    id : Object *
    out_ : IVertex *
    in_ : IVertex *
    label : string -> unit
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract.ValidateCreateEdge

e(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract.ValidateCreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract.ValidateCreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementFactoryContract.ValidateCreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

See Also

[ElementFactoryContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig Class

Configure how the GraphSON utility treats edge and vertex properties.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ElementPropertyConfig
```

VB

```
Public Class ElementPropertyConfig
```

C++


```
public ref class ElementPropertyConfig
```

F#





```
type ElementPropertyConfig = class end
```

The **ElementPropertyConfig** type exposes the following members.



Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------|
|  | ElementPropertyConfig | Initializes a new instance of the ElementPropertyConfig class |


Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | EdgePropertiesRule | |
|  | EdgePropertyKeys | |
|  | VertexPropertiesRule | |
|  | VertexPropertyKeys | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------|
|  | ExcludeProperties | Construct a configuration that excludes the specified properties from both vertices and edges. |
|  | IncludeProperties | Construct a configuration that includes the specified properties from both vertices and edges. |

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------|
|  | AllProperties | A configuration that includes all properties of vertices and edges. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig Constructor

Initializes a new instance of the [ElementPropertyConfig](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ElementPropertyConfig(
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys,
    ElementPropertyConfig.ElementPropertiesRule vertexPropertiesRule,
    ElementPropertyConfig.ElementPropertiesRule edgePropertiesRule
)
```

VB

```
Public Sub New (
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String),
    vertexPropertiesRule As ElementPropertyConfig.ElementPropertiesRule,
    edgePropertiesRule As ElementPropertyConfig.ElementPropertiesRule
)
```

C++

```
public:
ElementPropertyConfig(
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys,
    ElementPropertyConfig.ElementPropertiesRule vertexPropertiesRule,
    ElementPropertyConfig.ElementPropertiesRule edgePropertiesRule
)
```

F#

```
new :
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> *
    vertexPropertiesRule : ElementPropertyConfig.ElementPropertiesRule *
    edgePropertiesRule : ElementPropertyConfig.ElementPropertiesRule ->
ElementPropertyConfig
```

Parameters

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable<String>](#)

[Missing <param name="vertexPropertyKeys"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.#ctor(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String},VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule,

**VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule
)"]**

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

**[Missing <param name="edgePropertyKeys"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.#ctor(System.Colle
ctions.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String},V
elocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule,
VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule
)"]**

vertexPropertiesRule

Type:

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule](#)

**[Missing <param name="vertexPropertiesRule"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.#ctor(System.Colle
ctions.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String},V
elocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule,
VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule
)"]**

edgePropertiesRule

Type:

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule](#)

**[Missing <param name="edgePropertiesRule"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.#ctor(System.Colle
ctions.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String},V
elocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule,
VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule
)"]**

See Also





[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.ElementPropertyConfig Properties

The [ElementPropertyConfig](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------|-------------|
|  | EdgePropertiesRule | |
|  | EdgePropertyKeys | |
|  | VertexPropertiesRule | |
|  | VertexPropertyKeys | |

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.EdgePropertiesRule Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.EdgePropertiesRule"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ElementPropertyConfig.ElementPropertiesRule EdgePropertiesRule { get; }
}
```

VB

```
Public ReadOnly Property EdgePropertiesRule As
ElementPropertyConfig.ElementPropertiesRule
    Get
```

C++

```
public:
property ElementPropertyConfig.ElementPropertiesRule EdgePropertiesRule {
    ElementPropertyConfig.ElementPropertiesRule get ();
}
```

F#

```
member EdgePropertiesRule : ElementPropertyConfig.ElementPropertiesRule with
get
```

Property Value

Type: [ElementPropertyConfig.ElementPropertiesRule](#)

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.EdgePropertyKeys Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.EdgePropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> EdgePropertyKeys { get; }
```

VB

```
Public ReadOnly Property EdgePropertyKeys As IEnumerable(Of String)  
    Get
```

C++

```
public:  
property IEnumerable<String^>^ EdgePropertyKeys {  
    IEnumerable<String^>^ get ();  
}
```

F#

```
member EdgePropertyKeys : IEnumerable<string> with get
```

Property Value

Type: [IEnumerable\(String\)](#)

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.VertexPropertiesRule Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.VertexPropertiesRule"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ElementPropertyConfig.ElementPropertiesRule VertexPropertiesRule {  
    get; }  
}
```

VB

```
Public ReadOnly Property VertexPropertiesRule As  
    ElementPropertyConfig.ElementPropertiesRule  
    Get
```

C++

```
public:  
property ElementPropertyConfig.ElementPropertiesRule VertexPropertiesRule {  
    ElementPropertyConfig.ElementPropertiesRule get ();  
}
```

F#

```
member VertexPropertiesRule : ElementPropertyConfig.ElementPropertiesRule  
with get
```

Property Value

Type: [ElementPropertyConfig.ElementPropertiesRule](#)

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.VertexPropertyKeys Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.VertexPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> VertexPropertyKeys { get; }
```

VB

```
Public ReadOnly Property VertexPropertyKeys As IEnumerable(Of String)  
    Get
```

C++

```
public:  
property IEnumerable<String^>^ VertexPropertyKeys {  
    IEnumerable<String^>^ get ();  
}
```

F#

```
member VertexPropertyKeys : IEnumerable<string> with get
```

Property Value

Type: [IEnumerable\(String\)](#)

See Also





[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.ElementPropertyConfig Methods

The [ElementPropertyConfig](#) type exposes the following members.

Methods

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------|
|   | ExcludeProperties | Construct a configuration that excludes the specified properties from both vertices and edges. |
|   | IncludeProperties | Construct a configuration that includes the specified properties from both vertices and edges. |

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.ExcludeProperties Method

Construct a configuration that excludes the specified properties from both vertices and edges.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ElementPropertyConfig ExcludeProperties (
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys
)
```

VB

```
Public Shared Function ExcludeProperties (
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String)
) As ElementPropertyConfig
```

C++

```
public:
static ElementPropertyConfig^ ExcludeProperties (
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys
)
```

F#

```
static member ExcludeProperties :
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> -> ElementPropertyConfig
```

Parameters

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="vertexPropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ExcludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="edgePropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ExcludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

Return Value

Type: [ElementPropertyConfig](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ExcludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.IncludeProperties Method

Construct a configuration that includes the specified properties from both vertices and edges.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static ElementPropertyConfig IncludeProperties (
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys
)
```

VB

```
Public Shared Function IncludeProperties (
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String)
) As ElementPropertyConfig
```

C++

```
public:
static ElementPropertyConfig^ IncludeProperties (
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys
)
```

F#

```
static member IncludeProperties :
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> -> ElementPropertyConfig
```

Parameters

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="vertexPropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.IncludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="edgePropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.IncludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

Return Value

Type: [ElementPropertyConfig](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.IncludeProperties(System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

See Also


[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.ElementPropertyConfig Fields

The [ElementPropertyConfig](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------|
|  | AllProperties | A configuration that includes all properties of vertices and edges. |

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.AllProperties Field

A configuration that includes all properties of vertices and edges.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static readonly ElementPropertyConfig AllProperties
```

VB

```
Public Shared ReadOnly AllProperties As ElementPropertyConfig
```

C++

```
public:  
static initonly ElementPropertyConfig^ AllProperties
```

F#

```
static val AllProperties: ElementPropertyConfig
```

Field Value

Type: [ElementPropertyConfig](#)

See Also

[ElementPropertyConfig Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

ElementPropertyConfig.ElementPropertiesRule Enumeration

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig.ElementPropertiesRule"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum ElementPropertiesRule
```

VB

```
Public Enumeration ElementPropertiesRule
```

C++

```
public enum class ElementPropertiesRule
```

F#

```
type ElementPropertiesRule
```

Members

| | Member name | Value | Description |
|--|----------------|-------|-------------|
| | Include | 0 | |
| | Exclude | 1 | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphElementFactory Class

The standard factory used for most graph element creation. It uses an actual Graph implementation to construct vertices and edges

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphElementFactory : IElementFactory
```

VB

```
Public Class GraphElementFactory
    Implements IElementFactory
```

C++


```
public ref class GraphElementFactory : IElementFactory
```

F#

```
type GraphElementFactory =
    class
        interface IElementFactory
    end
```

The **GraphElementFactory** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------|
|  | GraphElementFactory | Initializes a new instance of the GraphElementFactory class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | CreateEdge | |
|  | CreateVertex | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphElementFactory Constructor

Initializes a new instance of the [GraphElementFactory](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphElementFactory(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GraphElementFactory(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GraphElementFactory
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also

[GraphElementFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphElementFactory.GraphElementFactory Methods

The [GraphElementFactory](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | CreateEdge | |
|  | CreateVertex | |

See Also

[GraphElementFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphElementFactory.CreateEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge CreateEdge(  
    Object id,  
    IVertex out_,  
    IVertex in_,  
    string label  
)
```

VB

```
Public Function CreateEdge (  
    id As Object,  
    out_ As IVertex,  
    in_ As IVertex,  
    label As String  
) As IEdge
```

C++

```
public:  
virtual IEdge^ CreateEdge(  
    Object^ id,  
    IVertex^ out_,  
    IVertex^ in_,  
    String^ label  
) sealed
```

F#

```
abstract CreateEdge :  
    id : Object *  
    out_ : IVertex *  
    in_ : IVertex *  
    label : string -> IEdge  
override CreateEdge :  
    id : Object *  
    out_ : IVertex *  
    in_ : IVertex *  
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IElementFactory.CreateEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[GraphElementFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphElementFactory.CreateVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex CreateVertex(  
    Object id  
)
```

VB

```
Public Function CreateVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ CreateVertex(  
    Object^ id  
) sealed
```

F#

```
abstract CreateVertex :  
    id : Object -> IVertex  
override CreateVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphElementFactory.CreateVertex(System.Object)"]

VelocityDB Class Library

Implements

[IElementFactory.CreateVertex\(Object\)](#)

See Also

[GraphElementFactory Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonMode Enumeration

Modes of operation of the GraphSONUtility.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum GraphSonMode
```

VB

```
Public Enumeration GraphSonMode
```

C++

```
public enum class GraphSonMode
```

F#

```
type GraphSonMode
```

Members

| Member name | Value | Description |
|-----------------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| COMPACT | 0 | COMPACT constructs GraphSON on the assumption that all property keys are fair game for exclusion including <code>_type</code> , <code>_inV</code> , <code>_outV</code> , <code>_label</code> and <code>_id</code> . It is possible to write GraphSON that cannot be read back into Graph, if some or all of these keys are excluded. |
| NORMAL | 1 | NORMAL includes the <code>_type</code> field and JSON data typing. |
| EXTENDED | 2 | EXTENDED includes the <code>_type</code> field and explicit data typing. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader Class

GraphSONReader reads the data from a TinkerPop JSON stream to a graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphSonReader
```

VB

```
Public Class GraphSonReader
```

C++


```
public ref class GraphSonReader
```

F#





```
type GraphSonReader = class end
```













The **GraphSonReader** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | GraphSonReader | Initializes a new instance of the GraphSonReader class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|   | InputGraph(IGraph, Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32) | |
|   | InputGraph(IGraph, String, Int32) | |
|   | InputGraph(IGraph, Stream, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader Constructor

Initializes a new instance of the [GraphSonReader](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphSonReader (  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GraphSonReader (  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GraphSonReader
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the JSON data

















See Also

[GraphSonReader Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.GraphSonReader Methods

Methods

















| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32) | |
|   | InputGraph(IGraph, String, Int32) | |
|   | InputGraph(IGraph, Stream, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[GraphSonReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method

Overload List

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|  | InputGraph(Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(Stream, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|  | InputGraph(String, Int32) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, String) | Input the JSON stream data into the graph. In practice, usually the provided graph is empty. |
|   | InputGraph(IGraph, Stream, Int32) | |
|   | InputGraph(IGraph, String, Int32) | |
|   | InputGraph(IGraph, Stream, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |
|   | InputGraph(IGraph, String, Int32, IEnumerable(String), IEnumerable(String)) | Input the JSON stream data into the graph. More control over how data is streamed is provided by this method. |

See Also

[GraphSonReader Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (Stream)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    Stream jsonInputStream  
)
```

VB

```
Public Sub InputGraph (  
    jsonInputStream As Stream  
)
```

C++

```
public:  
void InputGraph(  
    Stream^ jsonInputStream  
)
```

F#

```
member InputGraph :  
    jsonInputStream : Stream -> unit
```

Parameters

jsonInputStream

Type: [System.IO.Stream](#)

a Stream of JSON data

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (String)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename  
)
```

VB

```
Public Sub InputGraph (  
    filename As String  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename  
)
```

F#

```
member InputGraph :  
    filename : string -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file of JSON data

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (Stream, Int32)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    Stream jsonInputStream,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    jsonInputStream As Stream,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    Stream^ jsonInputStream,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    jsonInputStream : Stream *  
    bufferSize : int -> unit
```

Parameters

jsonInputStream

Type: [System.IO.Stream](#)

an Stream of JSON data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (String, Int32)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void InputGraph(  
    string filename,  
    int bufferSize  
)
```

VB

```
Public Sub InputGraph (  
    filename As String,  
    bufferSize As Integer  
)
```

C++

```
public:  
void InputGraph(  
    String^ filename,  
    int bufferSize  
)
```

F#

```
member InputGraph :  
    filename : string *  
    bufferSize : int -> unit
```

Parameters

filename

Type: [System.String](#)

name of a file of JSON data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, Stream)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph graph,  
    Stream jsonInputStream  
)
```

VB

```
Public Shared Sub InputGraph (  
    graph As IGraph,  
    jsonInputStream As Stream  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ graph,  
    Stream^ jsonInputStream  
)
```

F#

```
static member InputGraph :  
    graph : IGraph *  
    jsonInputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the JSON data

jsonInputStream

Type: [System.IO.Stream](#)

a Stream of JSON data

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, String)

Input the JSON stream data into the graph. In practice, usually the provided graph is empty.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub InputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member InputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the JSON data

filename

Type: [System.String](#)

name of a file of JSON data

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, Stream, Int32)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream jsonInputStream,  
    int bufferSize  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    jsonInputStream As Stream,  
    bufferSize As Integer  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ jsonInputStream,  
    int bufferSize  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    jsonInputStream : Stream *  
    bufferSize : int -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="inputGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32)"]

jsonInputStream

Type: [System.IO.Stream](#)

[Missing <param name="jsonInputStream"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32)"]

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32)"]

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, String, Int32)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename,  
    int bufferSize  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String,  
    bufferSize As Integer  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename,  
    int bufferSize  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string *  
    bufferSize : int -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="inputGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

filename

Type: [System.String](#)

[Missing <param name="filename"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

bufferSize

Type: [System.Int32](#)

[Missing <param name="bufferSize"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32)"]

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, Stream, Int32, IEnumerable(String), IEnumerable(String))

Input the JSON stream data into the graph. More control over how data is streamed is provided by this method.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    Stream jsonInputStream,  
    int bufferSize,  
    IEnumerable<string> edgePropertyKeys,  
    IEnumerable<string> vertexPropertyKeys  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    jsonInputStream As Stream,  
    bufferSize As Integer,  
    edgePropertyKeys As IEnumerable(Of String),  
    vertexPropertyKeys As IEnumerable(Of String)  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    Stream^ jsonInputStream,  
    int bufferSize,  
    IEnumerable<String^>^ edgePropertyKeys,  
    IEnumerable<String^>^ vertexPropertyKeys  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    jsonInputStream : Stream *  
    bufferSize : int *  
    edgePropertyKeys : IEnumerable<string> *  
    vertexPropertyKeys : IEnumerable<string> -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the JSON data

jsonInputStream

Type: [System.IO.Stream](#)

a Stream of JSON data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="edgePropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="vertexPropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.IO.Stream,System.Int32,System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonReader.InputGraph Method (IGraph, String, Int32, IEnumerable(String), IEnumerable(String))

Input the JSON stream data into the graph. More control over how data is streamed is provided by this method.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void InputGraph(  
    IGraph inputGraph,  
    string filename,  
    int bufferSize,  
    IEnumerable<string> edgePropertyKeys,  
    IEnumerable<string> vertexPropertyKeys  
)
```

VB

```
Public Shared Sub InputGraph (  
    inputGraph As IGraph,  
    filename As String,  
    bufferSize As Integer,  
    edgePropertyKeys As IEnumerable(Of String),  
    vertexPropertyKeys As IEnumerable(Of String)  
)
```

C++

```
public:  
static void InputGraph(  
    IGraph^ inputGraph,  
    String^ filename,  
    int bufferSize,  
    IEnumerable<String^>^ edgePropertyKeys,  
    IEnumerable<String^>^ vertexPropertyKeys  
)
```

F#

```
static member InputGraph :  
    inputGraph : IGraph *  
    filename : string *  
    bufferSize : int *  
    edgePropertyKeys : IEnumerable<string> *  
    vertexPropertyKeys : IEnumerable<string> -> unit
```

Parameters

inputGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to populate with the JSON data

filename

Type: [System.String](#)

name of a file of JSON data

bufferSize

Type: [System.Int32](#)

the amount of elements to hold in memory before committing a transactions (only valid for TransactionalGraphs)

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="edgePropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32,System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="vertexPropertyKeys"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonReader.InputGraph(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.Int32,System.Collections.Generic.IEnumerable{System.String},System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonReader Class](#)

[InputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphSonTokens
```

VB

```
Public NotInheritable Class GraphSonTokens
```

C++





















```
public ref class GraphSonTokens abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphSonTokens = class end
```

The **GraphSonTokens** type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|-------------|
|   | Edge | |
|   | Edges | |
|   | Id | |
|   | InV | |
|   | Label | |
|   | Mode | |
|   | OutV | |
|   | Type | |
|   | TypeBoolean | |
|   | TypeDouble | |

VelocityDB Class Library

| | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|--|
|   | TypeFloat | |
|   | TypeInteger | |
|   | TypeList | |
|   | TypeLong | |
|   | TypeMap | |
|   | TypeString | |
|   | TypeUnknown | |
|   | UnderscoreType | |
|   | Value | |
|   | Vertex | |
|   | Vertices | |






















See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.GraphSonTokens Fields

The [GraphSonTokens](#) type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Edge | |
|  | Edges | |
|  | Id | |
|  | InV | |
|  | Label | |
|  | Mode | |
|  | OutV | |
|  | Type | |
|  | TypeBoolean | |
|  | TypeDouble | |
|  | TypeFloat | |
|  | TypeInteger | |
|  | TypeList | |
|  | TypeLong | |
|  | TypeMap | |
|  | TypeString | |
|  | TypeUnknown | |
|  | UnderscoreType | |
|  | Value | |
|  | Vertex | |
|  | Vertices | |

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Edge Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Edge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Edge = "edge"
```

VB

```
Public Const Edge As String = "edge"
```

C++

```
public:  
literal String^ Edge = "edge"
```

F#

```
static val mutable Edge: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Edges Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Edges = "edges"
```

VB

```
Public Const Edges As String = "edges"
```

C++

```
public:  
literal String^ Edges = "edges"
```

F#

```
static val mutable Edges: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Id Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Id = "_id"
```

VB

```
Public Const Id As String = "_id"
```

C++

```
public:  
literal String^ Id = "_id"
```

F#

```
static val mutable Id: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.InV Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.InV"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string InV = "_inV"
```

VB

```
Public Const InV As String = "_inV"
```

C++

```
public:  
literal String^ InV = "_inV"
```

F#

```
static val mutable InV: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Label Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Label = "_label"
```

VB

```
Public Const Label As String = "_label"
```

C++

```
public:  
literal String^ Label = "_label"
```

F#

```
static val mutable Label: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Mode Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Mode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Mode = "mode"
```

VB

```
Public Const Mode As String = "mode"
```

C++

```
public:  
literal String^ Mode = "mode"
```

F#

```
static val mutable Mode: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.OutV Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.OutV"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string OutV = "_outV"
```

VB

```
Public Const OutV As String = "_outV"
```

C++

```
public:  
literal String^ OutV = "_outV"
```

F#

```
static val mutable OutV: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Type Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Type = "type"
```

VB

```
Public Const Type As String = "type"
```

C++

```
public:  
literal String^ Type = "type"
```

F#

```
static val mutable Type: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeBoolean Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeBoolean"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeBoolean = "boolean"
```

VB

```
Public Const TypeBoolean As String = "boolean"
```

C++

```
public:  
literal String^ TypeBoolean = "boolean"
```

F#

```
static val mutable TypeBoolean: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeDouble Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeDouble"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeDouble = "double"
```

VB

```
Public Const TypeDouble As String = "double"
```

C++

```
public:  
literal String^ TypeDouble = "double"
```

F#

```
static val mutable TypeDouble: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeFloat Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeFloat"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeFloat = "float"
```

VB

```
Public Const TypeFloat As String = "float"
```

C++

```
public:  
literal String^ TypeFloat = "float"
```

F#

```
static val mutable TypeFloat: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeInteger Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeInteger"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeInteger = "integer"
```

VB

```
Public Const TypeInteger As String = "integer"
```

C++

```
public:  
literal String^ TypeInteger = "integer"
```

F#

```
static val mutable TypeInteger: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeList Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeList"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeList = "list"
```

VB

```
Public Const TypeList As String = "list"
```

C++

```
public:  
literal String^ TypeList = "list"
```

F#

```
static val mutable TypeList: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeLong Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeLong"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeLong = "long"
```

VB

```
Public Const TypeLong As String = "long"
```

C++

```
public:  
literal String^ TypeLong = "long"
```

F#

```
static val mutable TypeLong: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeMap Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeMap"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeMap = "map"
```

VB

```
Public Const TypeMap As String = "map"
```

C++

```
public:  
literal String^ TypeMap = "map"
```

F#

```
static val mutable TypeMap: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeString Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeString = "string"
```

VB

```
Public Const TypeString As String = "string"
```

C++

```
public:  
literal String^ TypeString = "string"
```

F#

```
static val mutable TypeString: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.TypeUnknown Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.TypeUnknown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string TypeUnknown = "unknown"
```

VB

```
Public Const TypeUnknown As String = "unknown"
```

C++

```
public:  
literal String^ TypeUnknown = "unknown"
```

F#

```
static val mutable TypeUnknown: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.UnderscoreType Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.UnderscoreType"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string UnderscoreType = "_type"
```

VB

```
Public Const UnderscoreType As String = "_type"
```

C++

```
public:  
literal String^ UnderscoreType = "_type"
```

F#

```
static val mutable UnderscoreType: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Value Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Value"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Value = "value"
```

VB

```
Public Const Value As String = "value"
```

C++

```
public:  
literal String^ Value = "value"
```

F#

```
static val mutable Value: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Vertex Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Vertex"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Vertex = "vertex"
```

VB

```
Public Const Vertex As String = "vertex"
```

C++

```
public:  
literal String^ Vertex = "vertex"
```

F#

```
static val mutable Vertex: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonTokens.Vertices Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonTokens.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Vertices = "vertices"
```

VB

```
Public Const Vertices As String = "vertices"
```

C++

```
public:  
literal String^ Vertices = "vertices"
```

F#

```
static val mutable Vertices: string
```

Field Value

Type: [String](#)

See Also

[GraphSonTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility Class

Helps write individual graph elements to TinkerPop JSON format known as GraphSON.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphSonUtility
```

VB

```
Public Class GraphSonUtility
```

C++

```
public ref class GraphSonUtility
```

F#

```
type GraphSonUtility = class end
```














The **GraphSonUtility** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
|  | GraphSonUtility(GraphSonMode, IElementFactory) | A GraphSONUtility that includes all properties of vertices and edges. |
|  | GraphSonUtility(GraphSonMode, IElementFactory, ElementPropertyConfig) | Initializes a new instance of the GraphSonUtility class |
|  | GraphSonUtility(GraphSonMode, IElementFactory, IEnumerable(String), IEnumerable(String)) | A GraphSONUtility that includes the specified properties. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------|
|  | EdgeFromJson(JObject, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|  | EdgeFromJson(Stream, IVertex, IVertex) | |




| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
|  EdgeFromJson(String, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|  EdgeFromJson(JObject, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|  EdgeFromJson(Stream, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|  EdgeFromJson(String, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|  GetTypedValueFromJsonNode | |
|  JsonFromElement(IElement) | Creates GraphSON for a single graph element. |
|  JsonFromElement(IElement, IEnumerable(String), GraphSonMode) | Creates a JSON.NET ObjectNode from a graph element. |
|  VertexFromJson(JObject) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  VertexFromJson(Stream) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  VertexFromJson(String) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  VertexFromJson(JObject, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|  VertexFromJson(Stream, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|  VertexFromJson(String, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
|  | GraphSonUtility(GraphSonMode, IElementFactory) | A GraphSONUtility that includes all properties of vertices and edges. |
|  | GraphSonUtility(GraphSonMode, IElementFactory, ElementPropertyConfig) | Initializes a new instance of the GraphSonUtility class |
|  | GraphSonUtility(GraphSonMode, IElementFactory, IEnumerable(String), IEnumerable(String)) | A GraphSONUtility that includes the specified properties. |

See Also

[GraphSonUtility Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility Constructor (GraphSonMode, IElementFactory)

A GraphSONUtility that includes all properties of vertices and edges.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory factory  
)
```

VB

```
Public Sub New (  
    mode As GraphSonMode,  
    factory As IElementFactory  
)
```

C++

```
public:  
GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory^ factory  
)
```

F#

```
new :  
    mode : GraphSonMode *  
    factory : IElementFactory -> GraphSonUtility
```

Parameters

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

[Missing <param name="mode"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory)"]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

[Missing <param name="factory"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory)"]

VelocityDB Class Library

See Also

[GraphSonUtility Class](#)

[GraphSonUtility Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility Constructor (GraphSonMode, IElementFactory, ElementPropertyConfig)

Initializes a new instance of the [GraphSonUtility](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory factory,  
    ElementPropertyConfig config  
)
```

VB

```
Public Sub New (  
    mode As GraphSonMode,  
    factory As IElementFactory,  
    config As ElementPropertyConfig  
)
```

C++

```
public:  
GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory^ factory,  
    ElementPropertyConfig^ config  
)
```

F#

```
new :  
    mode : GraphSonMode *  
    factory : IElementFactory *  
    config : ElementPropertyConfig -> GraphSonUtility
```

Parameters

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

[Missing <param name="mode"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig)"]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

**[Missing <param name="factory"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Front
enac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphS
ON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig)"]**

config

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig](#)

**[Missing <param name="config"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Front
enac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphS
ON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.ElementPropertyConfig)"]**

See Also

[GraphSonUtility Class](#)

[GraphSonUtility Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility Constructor (GraphSonMode, IElementFactory, IEnumerable(String), IEnumerable(String))

A GraphSONUtility that includes the specified properties.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory factory,  
    IEnumerable<string> vertexPropertyKeys,  
    IEnumerable<string> edgePropertyKeys  
)
```

VB

```
Public Sub New (  
    mode As GraphSonMode,  
    factory As IElementFactory,  
    vertexPropertyKeys As IEnumerable(Of String),  
    edgePropertyKeys As IEnumerable(Of String)  
)
```

C++

```
public:  
GraphSonUtility(  
    GraphSonMode mode,  
    IElementFactory^ factory,  
    IEnumerable<String^>^ vertexPropertyKeys,  
    IEnumerable<String^>^ edgePropertyKeys  
)
```

F#

```
new :  
    mode : GraphSonMode *  
    factory : IElementFactory *  
    vertexPropertyKeys : IEnumerable<string> *  
    edgePropertyKeys : IEnumerable<string> -> GraphSonUtility
```

Parameters

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

[Missing <param name="mode"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphS

ON.IElementFactory, System.Collections.Generic.IEnumerable{System.String}, System.Collections.Generic.IEnumerable{System.String}]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

[Missing <param name="factory"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode, VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory, System.Collections.Generic.IEnumerable{System.String}, System.Collections.Generic.IEnumerable{System.String})"]

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable{String}](#)

[Missing <param name="vertexPropertyKeys"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode, VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory, System.Collections.Generic.IEnumerable{System.String}, System.Collections.Generic.IEnumerable{System.String})"]

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable{String}](#)

[Missing <param name="edgePropertyKeys"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.#ctor(VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode, VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory, System.Collections.Generic.IEnumerable{System.String}, System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)
























[GraphSonUtility Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.GraphSonUtility Methods

The [GraphSonUtility](#) type exposes the following members.

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
|  | EdgeFromJson(JObject, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|  | EdgeFromJson(Stream, IVertex, IVertex) | |
|  | EdgeFromJson(String, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|   | EdgeFromJson(JObject, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|   | EdgeFromJson(Stream, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|   | EdgeFromJson(String, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|   | GetTypedValueFromJsonNode | |
|  | JsonFromElement(IElement) | Creates GraphSON for a single graph element. |
|   | JsonFromElement(IElement, IEnumerable(String), GraphSonMode) | Creates a JSON.NET ObjectNode from a graph element. |
|  | VertexFromJson(JObject) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  | VertexFromJson(Stream) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  | VertexFromJson(String) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|   | VertexFromJson(JObject, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|   | VertexFromJson(Stream, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|   | VertexFromJson(String, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |

VelocityDB Class Library










See Also

[GraphSonUtility Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  | EdgeFromJson(JObject, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|  | EdgeFromJson(Stream, IVertex, IVertex) | |
|  | EdgeFromJson(String, IVertex, IVertex) | Creates an edge from GraphSON using settings supplied in the constructor. |
|   | EdgeFromJson(JObject, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|   | EdgeFromJson(Stream, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |
|   | EdgeFromJson(String, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Edge from JSON. The edge must match the accepted GraphSON format. |

See Also

[GraphSonUtility Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (JObject, IVertex, IVertex)

Creates an edge from GraphSON using settings supplied in the constructor.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge EdgeFromJson(  
    JObject json,  
    IVertex out_,  
    IVertex in_  
)
```

VB

```
Public Function EdgeFromJson (  
    json As JObject,  
    out_ As IVertex,  
    in_ As IVertex  
) As IEdge
```

C++

```
public:  
IEdge^ EdgeFromJson(  
    JObject^ json,  
    IVertex^ out_,  
    IVertex^ in_  
)
```

F#

```
member EdgeFromJson :  
    json : JObject *  
    out_ : IVertex *  
    in_ : IVertex -> IEdge
```

Parameters

json

Type: **JObject**

[Missing <param name="json"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (Stream, IVertex, IVertex)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge EdgeFromJson(  
    Stream json,  
    IVertex out_,  
    IVertex in_  
)
```

VB

```
Public Function EdgeFromJson (  
    json As Stream,  
    out_ As IVertex,  
    in_ As IVertex  
) As IEdge
```

C++

```
public:  
IEdge^ EdgeFromJson(  
    Stream^ json,  
    IVertex^ out_,  
    IVertex^ in_  
)
```

F#

```
member EdgeFromJson :  
    json : Stream *  
    out_ : IVertex *  
    in_ : IVertex -> IEdge
```

Parameters

json

Type: [System.IO.Stream](#)

[Missing <param name="json"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

**[Missing <param name="out_" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.S
tream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]**

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

**[Missing <param name="in_" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.S
tream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]**

Return Value

Type: [IEdge](#)

**[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.S
tream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]**

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (String, IVertex, IVertex)

Creates an edge from GraphSON using settings supplied in the constructor.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge EdgeFromJson(  
    string json,  
    IVertex out_,  
    IVertex in_  
)
```

VB

```
Public Function EdgeFromJson (  
    json As String,  
    out_ As IVertex,  
    in_ As IVertex  
) As IEdge
```

C++

```
public:  
IEdge^ EdgeFromJson(  
    String^ json,  
    IVertex^ out_,  
    IVertex^ in_  
)
```

F#

```
member EdgeFromJson :  
    json : string *  
    out_ : IVertex *  
    in_ : IVertex -> IEdge
```

Parameters

json

Type: [System.String](#)

[Missing <param name="json"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (JObject, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Edge from JSON. The edge must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEdge EdgeFromJson(
    JObject json,
    IVertex out_,
    IVertex in_,
    IElementFactory factory,
    GraphSonMode mode,
    IEnumerable<string> propertyKeys
)
```

VB

```
Public Shared Function EdgeFromJson (
    json As JObject,
    out_ As IVertex,
    in_ As IVertex,
    factory As IElementFactory,
    mode As GraphSonMode,
    propertyKeys As IEnumerable(Of String)
) As IEdge
```

C++

```
public:
static IEdge^ EdgeFromJson(
    JObject^ json,
    IVertex^ out_,
    IVertex^ in_,
    IElementFactory^ factory,
    GraphSonMode mode,
    IEnumerable<String^>^ propertyKeys
)
```

F#

```
static member EdgeFromJson :
    json : JObject *
    out_ : IVertex *
    in_ : IVertex *
    factory : IElementFactory *
    mode : GraphSonMode *
    propertyKeys : IEnumerable<string> -> IEdge
```

Parameters

json

Type: **JsonObject**

a single edge in GraphSON format as a Stream

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JsonObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JsonObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable{String}](#)

a list of keys to include when reading of element properties

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(Newtonsoft.Json.Linq.JsonObject,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (Stream, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Edge from JSON. The edge must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEdge EdgeFromJson(
    Stream json,
    IVertex out_,
    IVertex in_,
    IElementFactory factory,
    GraphSonMode mode,
    IEnumerable<string> propertyKeys
)
```

VB

```
Public Shared Function EdgeFromJson (
    json As Stream,
    out_ As IVertex,
    in_ As IVertex,
    factory As IElementFactory,
    mode As GraphSonMode,
    propertyKeys As IEnumerable(Of String)
) As IEdge
```

C++

```
public:
static IEdge^ EdgeFromJson(
    Stream^ json,
    IVertex^ out_,
    IVertex^ in_,
    IElementFactory^ factory,
    GraphSonMode mode,
    IEnumerable<String^>^ propertyKeys
)
```

F#

```
static member EdgeFromJson :
    json : Stream *
    out_ : IVertex *
    in_ : IVertex *
    factory : IElementFactory *
    mode : GraphSonMode *
    propertyKeys : IEnumerable<string> -> IEdge
```

Parameters

json

Type: [System.IO.Stream](#)

a single edge in GraphSON format as a Stream

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

a list of keys to include when reading of element properties

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.EdgeFromJson Method (String, IVertex, IVertex, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Edge from JSON. The edge must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IEdge EdgeFromJson(
    string json,
    IVertex out_,
    IVertex in_,
    IElementFactory factory,
    GraphSonMode mode,
    IEnumerable<string> propertyKeys
)
```

VB

```
Public Shared Function EdgeFromJson (
    json As String,
    out_ As IVertex,
    in_ As IVertex,
    factory As IElementFactory,
    mode As GraphSonMode,
    propertyKeys As IEnumerable(Of String)
) As IEdge
```

C++

```
public:
static IEdge^ EdgeFromJson(
    String^ json,
    IVertex^ out_,
    IVertex^ in_,
    IElementFactory^ factory,
    GraphSonMode mode,
    IEnumerable<String^>^ propertyKeys
)
```

F#

```
static member EdgeFromJson :
    json : string *
    out_ : IVertex *
    in_ : IVertex *
    factory : IElementFactory *
    mode : GraphSonMode *
    propertyKeys : IEnumerable<string> -> IEdge
```

Parameters

json

Type: [System.String](#)

a single edge in GraphSON format as a string

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable{String}](#)

a list of keys to include when reading of element properties

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.EdgeFromJson(System.String,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)

[EdgeFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.GetTypedValueFromJsonNode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.GetTypedValueFromJsonNode(Newtonsoft.Json.Linq.JToken)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static Object GetTypedValueFromJsonNode (
    JToken node
)
```

VB

```
Public Shared Function GetTypedValueFromJsonNode (
    node As JToken
) As Object
```

C++

```
public:
static Object^ GetTypedValueFromJsonNode (
    JToken^ node
)
```

F#

```
static member GetTypedValueFromJsonNode :
    node : JToken -> Object
```

Parameters

node

Type: **JToken**

[Missing <param name="node"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.GetTypedValueFromJsonNode(Newtonsoft.Json.Linq.JToken)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.GetTypedValueFromJsonNode(Newtonsoft.Json.Linq.JToken)"]




See Also

[GraphSonUtility Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.JsonFromElement Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------------------------------|
|  | JsonFromElement(IElement) | Creates GraphSON for a single graph element. |
|   | JsonFromElement(IElement, IEnumerable(String), GraphSonMode) | Creates a JSON.NET ObjectNode from a graph element. |

See Also

[GraphSonUtility Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.JsonFromElement Method (IElement)

Creates GraphSON for a single graph element.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public JObject JsonFromElement (
    IElement element
)
```

VB

```
Public Function JsonFromElement (
    element As IElement
) As JObject
```

C++

```
public:
JObject^ JsonFromElement (
    IElement^ element
)
```

F#

```
member JsonFromElement :
    element : IElement -> JObject
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.JsonFromElement(VelocityGraph.Frontenac.Blueprints.IElement)"]

Return Value

Type: **JObject**

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.JsonFromElement(VelocityGraph.Frontenac.Blueprints.IElement)"]

See Also

[GraphSonUtility Class](#)

[JsonFromElement Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.JsonFromElement Method (IElement, IEnumerable(String), GraphSonMode)

Creates a JSON.NET JObject from a graph element.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static JObject JsonFromElement (
    IElement element,
    IEnumerable<string> propertyKeys,
    GraphSonMode mode
)
```

VB

```
Public Shared Function JsonFromElement (
    element As IElement,
    propertyKeys As IEnumerable(Of String),
    mode As GraphSonMode
) As JObject
```

C++

```
public:
static JObject^ JsonFromElement (
    IElement^ element,
    IEnumerable<String^>^ propertyKeys,
    GraphSonMode mode
)
```

F#

```
static member JsonFromElement :
    element : IElement *
    propertyKeys : IEnumerable<string> *
    mode : GraphSonMode -> JObject
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

the graph element to convert to JSON.

propertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

The property keys at the root of the element to serialize. If null, then all keys are serialized.

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

The type of GraphSON to generate.

Return Value

Type: **JsonObject**

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.JsonFromElement(VelocityGraph.Frontenac.Blueprints.IElement,System.Collections.Generic.IEnumerable{System.String},VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode)"]

See Also










[GraphSonUtility Class](#)

[JsonFromElement Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method

Overload List

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
|  | VertexFromJson(JObject) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  | VertexFromJson(Stream) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|  | VertexFromJson(String) | Creates a vertex from GraphSON using settings supplied in the constructor. |
|   | VertexFromJson(JObject, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|   | VertexFromJson(Stream, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |
|   | VertexFromJson(String, IElementFactory, GraphSonMode, IEnumerable(String)) | Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format. |

See Also

[GraphSonUtility Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (JObject)

Creates a vertex from GraphSON using settings supplied in the constructor.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex VertexFromJson(  
    JObject json  
)
```

VB

```
Public Function VertexFromJson (  
    json As JObject  
) As IVertex
```

C++

```
public:  
IVertex^ VertexFromJson(  
    JObject^ json  
)
```

F#

```
member VertexFromJson :  
    json : JObject -> IVertex
```

Parameters

json

Type: **JObject**

[Missing <param name="json"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(Newtons of.Json.Linq.JObject)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(Newtons of.Json.Linq.JObject)"]

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (Stream)

Creates a vertex from GraphSON using settings supplied in the constructor.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex VertexFromJson(  
    Stream json  
)
```

VB

```
Public Function VertexFromJson (  
    json As Stream  
) As IVertex
```

C++

```
public:  
IVertex^ VertexFromJson(  
    Stream^ json  
)
```

F#

```
member VertexFromJson :  
    json : Stream -> IVertex
```

Parameters

json

Type: [System.IO.Stream](#)

[Missing <param name="json"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.IO.Stream)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.IO.Stream)"]

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (String)

Creates a vertex from GraphSON using settings supplied in the constructor.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex VertexFromJson(  
    string json  
)
```

VB

```
Public Function VertexFromJson (  
    json As String  
) As IVertex
```

C++

```
public:  
IVertex^ VertexFromJson(  
    String^ json  
)
```

F#

```
member VertexFromJson :  
    json : string -> IVertex
```

Parameters

json

Type: [System.String](#)

[Missing <param name="json"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.String)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.String)"]

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (JObject, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertex VertexFromJson (
    JObject json,
    IElementFactory factory,
    GraphSonMode mode,
    IEnumerable<string> propertyKeys
)
```

VB

```
Public Shared Function VertexFromJson (
    json As JObject,
    factory As IElementFactory,
    mode As GraphSonMode,
    propertyKeys As IEnumerable(Of String)
) As IVertex
```

C++

```
public:
static IVertex^ VertexFromJson (
    JObject^ json,
    IElementFactory^ factory,
    GraphSonMode mode,
    IEnumerable<String^>^ propertyKeys
)
```

F#

```
static member VertexFromJson :
    json : JObject *
    factory : IElementFactory *
    mode : GraphSonMode *
    propertyKeys : IEnumerable<string> -> IVertex
```

Parameters

json

Type: **JObject**

a single vertex in GraphSON format as Jackson JsonNode

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

a list of keys to include on reading of element properties

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(Newtonsoft.Json.Linq.JObject,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (Stream, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertex VertexFromJson (
    Stream json,
    IElementFactory factory,
    GraphSonMode mode,
    IEnumerable<string> propertyKeys
)
```

VB

```
Public Shared Function VertexFromJson (
    json As Stream,
    factory As IElementFactory,
    mode As GraphSonMode,
    propertyKeys As IEnumerable(Of String)
) As IVertex
```

C++

```
public:
static IVertex^ VertexFromJson (
    Stream^ json,
    IElementFactory^ factory,
    GraphSonMode mode,
    IEnumerable<String^>^ propertyKeys
)
```

F#

```
static member VertexFromJson :
    json : Stream *
    factory : IElementFactory *
    mode : GraphSonMode *
    propertyKeys : IEnumerable<string> -> IVertex
```

Parameters

json

Type: [System.IO.Stream](#)

a single vertex in GraphSON format as a Stream.

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

a list of keys to include on reading of element properties

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.IO.Stream,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonUtility.VertexFromJson Method (String, IElementFactory, GraphSonMode, IEnumerable(String))

Reads an individual Vertex from JSON. The vertex must match the accepted GraphSON format.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertex VertexFromJson(  
    string json,  
    IElementFactory factory,  
    GraphSonMode mode,  
    IEnumerable<string> propertyKeys  
)
```

VB

```
Public Shared Function VertexFromJson (  
    json As String,  
    factory As IElementFactory,  
    mode As GraphSonMode,  
    propertyKeys As IEnumerable(Of String)  
) As IVertex
```

C++

```
public:  
static IVertex^ VertexFromJson(  
    String^ json,  
    IElementFactory^ factory,  
    GraphSonMode mode,  
    IEnumerable<String^>^ propertyKeys  
)
```

F#

```
static member VertexFromJson :  
    json : string *  
    factory : IElementFactory *  
    mode : GraphSonMode *  
    propertyKeys : IEnumerable<string> -> IVertex
```

Parameters

json

Type: [System.String](#)

a single vertex in GraphSON format as a string.

factory

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory](#)

the factory responsible for constructing graph elements

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

the mode of the GraphSON

propertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

a list of keys to include on reading of element properties

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonUtility.VertexFromJson(System.String,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory,VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode,System.Collections.Generic.IEnumerable{System.String})"

See Also

[GraphSonUtility Class](#)

[VertexFromJson Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter Class

GraphSONWriter writes a Graph to a TinkerPop JSON OutputStream.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonWriter

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class GraphSonWriter
```

VB

```
Public Class GraphSonWriter
```

C++


```
public ref class GraphSonWriter
```

F#









```
type GraphSonWriter = class end
```

The **GraphSonWriter** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | GraphSonWriter | Initializes a new instance of the GraphSonWriter class |

Methods

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|   | OutputGraph(IGraph, Stream, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |
|   | OutputGraph(IGraph, String, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
|  OutputGraph(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|  OutputGraph(String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|   OutputGraph(IGraph, Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|   OutputGraph(IGraph, String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter Constructor

Initializes a new instance of the [GraphSonWriter](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public GraphSonWriter(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
GraphSonWriter(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> GraphSonWriter
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the Graph to pull the data from









See Also

[GraphSonWriter Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.GraphSonWriter Methods

Methods















| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|  | OutputGraph(IGraph, Stream) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|  | OutputGraph(IGraph, String) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|  | OutputGraph(IGraph, Stream, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |
|  | OutputGraph(IGraph, String, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |
|  | OutputGraph(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|  | OutputGraph(String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|  | OutputGraph(IGraph, Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|  | OutputGraph(IGraph, String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |

See Also

[GraphSonWriter Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method

Overload List

| | Name | Description |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|   | OutputGraph(IGraph, Stream) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|   | OutputGraph(IGraph, String) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL. |
|   | OutputGraph(IGraph, Stream, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |
|   | OutputGraph(IGraph, String, GraphSonMode) | Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. |
|  | OutputGraph(Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|  | OutputGraph(String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|   | OutputGraph(IGraph, Stream, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |
|   | OutputGraph(IGraph, String, IEnumerable(String), IEnumerable(String), GraphSonMode) | Write the data in a Graph to a JSON OutputStream. |

See Also

[GraphSonWriter Class](#)[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, Stream)

Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    Stream jsonOutputStream  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    jsonOutputStream As Stream  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    Stream^ jsonOutputStream  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    jsonOutputStream : Stream -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

jsonOutputStream

Type: [System.IO.Stream](#)

the JSON OutputStream to write the Graph data to

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, String)

Write the data in a Graph to a JSON OutputStream. All keys are written to JSON. Utilizing GraphSONMode.NORMAL.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

filename

Type: [System.String](#)

the JSON file to write the Graph data to

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, Stream, GraphSonMode)

Write the data in a Graph to a JSON OutputStream. All keys are written to JSON.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    Stream jsonOutputStream,
    GraphSonMode mode
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    jsonOutputStream As Stream,
    mode As GraphSonMode
)
```

C++

```
public:
static void OutputGraph (
    IGraph^ graph,
    Stream^ jsonOutputStream,
    GraphSonMode mode
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    jsonOutputStream : Stream *
    mode : GraphSonMode -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

jsonOutputStream

Type: [System.IO.Stream](#)

the JSON OutputStream to write the Graph data to

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)
determines the format of the GraphSON

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, String, GraphSonMode)

Write the data in a Graph to a JSON OutputStream. All keys are written to JSON.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph(  
    IGraph graph,  
    string filename,  
    GraphSonMode mode  
)
```

VB

```
Public Shared Sub OutputGraph (  
    graph As IGraph,  
    filename As String,  
    mode As GraphSonMode  
)
```

C++

```
public:  
static void OutputGraph(  
    IGraph^ graph,  
    String^ filename,  
    GraphSonMode mode  
)
```

F#

```
static member OutputGraph :  
    graph : IGraph *  
    filename : string *  
    mode : GraphSonMode -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

filename

Type: [System.String](#)

the JSON file to write the Graph data to

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

VelocityDB Class Library

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (Stream, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a JSON OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph (
    Stream jsonOutputStream,
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys,
    GraphSonMode mode
)
```

VB

```
Public Sub OutputGraph (
    jsonOutputStream As Stream,
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String),
    mode As GraphSonMode
)
```

C++

```
public:
void OutputGraph (
    Stream^ jsonOutputStream,
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys,
    GraphSonMode mode
)
```

F#

```
member OutputGraph :
    jsonOutputStream : Stream *
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> *
    mode : GraphSonMode -> unit
```

Parameters

jsonOutputStream

Type: [System.IO.Stream](#)

the JSON OutputStream to write the Graph data to

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

VelocityDB Class Library

the keys of the vertex elements to write to JSON

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (String, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a JSON OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void OutputGraph(  
    string filename,  
    IEnumerable<string> vertexPropertyKeys,  
    IEnumerable<string> edgePropertyKeys,  
    GraphSonMode mode  
)
```

VB

```
Public Sub OutputGraph (  
    filename As String,  
    vertexPropertyKeys As IEnumerable(Of String),  
    edgePropertyKeys As IEnumerable(Of String),  
    mode As GraphSonMode  
)
```

C++

```
public:  
void OutputGraph(  
    String^ filename,  
    IEnumerable<String^>^ vertexPropertyKeys,  
    IEnumerable<String^>^ edgePropertyKeys,  
    GraphSonMode mode  
)
```

F#

```
member OutputGraph :  
    filename : string *  
    vertexPropertyKeys : IEnumerable<string> *  
    edgePropertyKeys : IEnumerable<string> *  
    mode : GraphSonMode -> unit
```

Parameters

filename

Type: [System.String](#)

the JSON file to write the Graph data to

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

VelocityDB Class Library

the keys of the vertex elements to write to JSON

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, Stream, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a JSON OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    Stream jsonOutputStream,
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys,
    GraphSonMode mode
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    jsonOutputStream As Stream,
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String),
    mode As GraphSonMode
)
```

C++

```
public:
static void OutputGraph(
    IGraph^ graph,
    Stream^ jsonOutputStream,
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys,
    GraphSonMode mode
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    jsonOutputStream : Stream *
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> *
    mode : GraphSonMode -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

jsonOutputStream

Type: [System.IO.Stream](#)

the JSON OutputStream to write the Graph data to

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the vertex elements to write to JSON

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

GraphSonWriter.OutputGraph Method (IGraph, String, IEnumerable(String), IEnumerable(String), GraphSonMode)

Write the data in a Graph to a JSON OutputStream.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void OutputGraph (
    IGraph graph,
    string filename,
    IEnumerable<string> vertexPropertyKeys,
    IEnumerable<string> edgePropertyKeys,
    GraphSonMode mode
)
```

VB

```
Public Shared Sub OutputGraph (
    graph As IGraph,
    filename As String,
    vertexPropertyKeys As IEnumerable(Of String),
    edgePropertyKeys As IEnumerable(Of String),
    mode As GraphSonMode
)
```

C++

```
public:
static void OutputGraph(
    IGraph^ graph,
    String^ filename,
    IEnumerable<String^>^ vertexPropertyKeys,
    IEnumerable<String^>^ edgePropertyKeys,
    GraphSonMode mode
)
```

F#

```
static member OutputGraph :
    graph : IGraph *
    filename : string *
    vertexPropertyKeys : IEnumerable<string> *
    edgePropertyKeys : IEnumerable<string> *
    mode : GraphSonMode -> unit
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

the graph to serialize to JSON

filename

Type: [System.String](#)

the JSON file to write the Graph data to

vertexPropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the vertex elements to write to JSON

edgePropertyKeys

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

the keys of the edge elements to write to JSON

mode

Type: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.GraphSonMode](#)

determines the format of the GraphSON

See Also

[GraphSonWriter Class](#)

[OutputGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

IElementFactory Interface

A factory responsible for creating graph elements. Abstracts the way that graph elements are created. In most cases a Graph is responsible for element creation, but there are cases where more control over how vertices and edges are constructed.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IElementFactory
```

VB

```
Public Interface IElementFactory
```

C++



```
public interface class IElementFactory
```

F#

```
type IElementFactory = interface end
```

The **IElementFactory** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|--------------------------------|
|  | CreateEdge | Creates a new Edge instance. |
|  | CreateVertex | creates a new Vertex instance. |



See Also

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

IElementFactory.IElementFactory Methods

The [IElementFactory](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------|
|  | CreateEdge | Creates a new Edge instance. |
|  | CreateVertex | creates a new Vertex instance. |

See Also

[IElementFactory Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

IElementFactory.CreateEdge Method

Creates a new Edge instance.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IEdge CreateEdge(  
    Object id,  
    IVertex out_,  
    IVertex in_,  
    string label  
)
```

VB

```
Function CreateEdge (  
    id As Object,  
    out_ As IVertex,  
    in_ As IVertex,  
    label As String  
) As IEdge
```

C++

```
IEdge^ CreateEdge(  
    Object^ id,  
    IVertex^ out_,  
    IVertex^ in_,  
    String^ label  
)
```

F#

```
abstract CreateEdge :  
    id : Object *  
    out_ : IVertex *  
    in_ : IVertex *  
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

out_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="out_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

in_

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="in_" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label" /> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

See Also

[IElementFactory Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)

IElementFactory.CreateVertex Method

creates a new Vertex instance.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IVertex CreateVertex(  
    Object id  
)
```

VB

```
Function CreateVertex (  
    id As Object  
) As IVertex
```

C++

```
IVertex^ CreateVertex(  
    Object^ id  
)
```

F#

```
abstract CreateVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON.IElementFactory.CreateVertex(System.Object)"]

See Also




[IElementFactory Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.IO.GraphSON Namespace](#)


VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers"]

Classes

| Class | Description |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  WrappedQuery | |
|  WrapperGraphContract | |
|  WrapperVertexQuery | A WrapperQuery is useful for wrapping the construction and results of a Vertex.query(). Any necessary Iterable wrapping must occur when Vertex.vertices() or Vertex.edges() is called. |

Interfaces

| Interface | Description |
|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  IWrapperGraph | A WrapperGraph has an underlying graph object to which it delegates its operations. |

IWrapperGraph Interface

A WrapperGraph has an underlying graph object to which it delegates its operations.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IWrapperGraph
```

VB

```
Public Interface IWrapperGraph
```

C++


```
public interface class IWrapperGraph
```

F#

```
type IWrapperGraph = interface end
```

The **IWrapperGraph** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|------------------------------------------|
|  | GetBaseGraph | Get the graph this wrapper delegates to. |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

IWrapperGraph.IWrapperGraph Methods

The [IWrapperGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|------------------------------------------|
|  | GetBaseGraph | Get the graph this wrapper delegates to. |

See Also

[IWrapperGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

IWrapperGraph.GetBaseGraph Method

Get the graph this wrapper delegates to.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
IGraph GetBaseGraph ()
```

VB

```
Function GetBaseGraph As IGraph
```

C++

```
IGraph^ GetBaseGraph ()
```

F#

```
abstract GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

the underlying graph that this WrapperGraph delegates its operations to.

See Also

[IWrapperGraph Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class WrappedQuery : IQuery
```


```
VB
Public Class WrappedQuery
    Implements IQuery
```

```
C++
public ref class WrappedQuery : IQuery
```






```
F#
type WrappedQuery =
    class
        interface IQuery
    end
```

The **WrappedQuery** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | WrappedQuery | Initializes a new instance of the WrappedQuery class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Limit | |

| | | |
|-----------------------------------------------------------------------------------|--------------------------|--|
|  | Vertices | |
|-----------------------------------------------------------------------------------|--------------------------|--|

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery Constructor

Initializes a new instance of the [WrappedQuery](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedQuery(
    IQuery query,
    Func<IQuery, IEnumerable<IEdge>> edgesSelector,
    Func<IQuery, IEnumerable<IVertex>> verticesSelector
)
```

VB

```
Public Sub New (
    query As IQuery,
    edgesSelector As Func(Of IQuery, IEnumerable(Of IEdge)),
    verticesSelector As Func(Of IQuery, IEnumerable(Of IVertex))
)
```

C++

```
public:
    WrappedQuery(
        IQuery^ query,
        Func<IQuery^, IEnumerable<IEdge^>>^ edgesSelector,
        Func<IQuery^, IEnumerable<IVertex^>>^ verticesSelector
    )
```

F#

```
new :
    query : IQuery *
    edgesSelector : Func<IQuery, IEnumerable<IEdge>> *
    verticesSelector : Func<IQuery, IEnumerable<IVertex>> -> WrappedQuery
```

Parameters

query

Type: [VelocityGraph.Frontenac.Blueprints.IQuery](#)

[Missing <param name="query"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

edgesSelector

Type: [System.Func\(IQuery, IEnumerable\(IEdge\)\)](#)

[Missing <param name="edgesSelector"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

verticesSelector

Type: [System.Func\(IQuery, IEnumerable\(IVertex\)\)](#)

[Missing <param name="verticesSelector"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

See Also







[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.WrappedQuery Methods

The [WrappedQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Limit | |
|  | Vertices | |

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Edges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> Edges ()
```

VB

```
Public Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ Edges () sealed
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>  
override Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Edges"]

Implements

[IQuery.Edges\(\)](#)

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Has Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Has Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Has(  
    string key,  
    Object value  
)
```

VB

```
Public Function Has (  
    key As String,  
    value As Object  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Has(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Has :  
    key : string *  
    value : Object -> IQuery  
override Has :  
    key : string *  
    value : Object -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has(System.String,System.Object)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has(System.String,System.Object)"]

Implements

[IQuery.Has\(String, Object\)](#)

See Also

[WrappedQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Has(T) Method (String, Compare, T)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Has<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Public Function Has(Of T) (
    key As String,
    compare As Compare,
    value As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Has(
    String^ key,
    Compare compare,
    T value
) sealed
```

F#

```
abstract Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
override Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

value

Type: *T*

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Implements

[IQuery.Has\(T\)\(String, Compare, T\)](#)

See Also

[WrappedQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Interval(T) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Interval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Public Function Interval(Of T) (
    key As String,
    startValue As T,
    endValue As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Interval(
    String^ key,
    T startValue,
    T endValue
) sealed
```

F#

```
abstract Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
override Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

startValue

Type: **T**

[Missing <param name="startValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

endValue

Type: **T**

[Missing <param name="endValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Interval`1(System.String,`0,`0)"]

Implements

[IQuery.Interval\(T\)\(String, T, T\)](#)

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Limit Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Limit(System.Int64)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Limit(  
    long max  
)
```

VB

```
Public Function Limit (  
    max As Long  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Limit(  
    long long max  
) sealed
```

F#

```
abstract Limit :  
    max : int64 -> IQuery  
override Limit :  
    max : int64 -> IQuery
```

Parameters

max

Type: [System.Int64](#)

[Missing <param name="max"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Limit(System.Int64)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Limit(System.Int64)"]

Implements

[IQuery.Limit\(Int64\)](#)

VelocityDB Class Library

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrappedQuery.Vertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> Vertices ()
```

VB

```
Public Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ Vertices () sealed
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>  
override Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrappedQuery.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

See Also

[WrappedQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperGraphContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperGraphContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperGraphContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class WrapperGraphContract : IWrapperGraph
```

VB

```
Public MustInherit Class WrapperGraphContract  
    Implements IWrapperGraph
```

C++

```
public ref class WrapperGraphContract abstract : IWrapperGraph
```

F#

```
[<AbstractClassAttribute>]  
type WrapperGraphContract =  
    class  
        interface IWrapperGraph  
    end
```

The **WrapperGraphContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | GetBaseGraph | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperGraphContract.WrapperGraphContract Methods

The [WrapperGraphContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | GetBaseGraph | |

See Also

[WrapperGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperGraphContract.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperGraphContract.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperGraphContract.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[WrapperGraphContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery Class

A WrapperQuery is useful for wrapping the construction and results of a Vertex.query(). Any necessary Iterable wrapping must occur when Vertex.vertices() or Vertex.edges() is called.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrapperVertexQuery : IVertexQuery,
    IQuery
```

VB

```
Public Class WrapperVertexQuery
    Implements IVertexQuery, IQuery
```

C++


```
public ref class WrapperVertexQuery : IVertexQuery,
    IQuery
```

F#




```
type WrapperVertexQuery =
    class
        interface IVertexQuery
        interface IQuery
    end
```

The **WrapperVertexQuery** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------|
|  | WrapperVertexQuery | Initializes a new instance of the WrapperVertexQuery class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | Count | |
|  | Direction | |
|  | Edges | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|--------------------------------------------|--|
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Labels | |
|  | Limit | |
|  | VertexIds | |
|  | Vertices | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery Constructor

Initializes a new instance of the [WrapperVertexQuery](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrapperVertexQuery(
    IVertexQuery query,
    Func<IVertexQuery, IEnumerable<IEdge>> edgesSelector,
    Func<IVertexQuery, IEnumerable<IVertex>> verticesSelector
)
```

VB

```
Public Sub New (
    query As IVertexQuery,
    edgesSelector As Func(Of IVertexQuery, IEnumerable(Of IEdge)),
    verticesSelector As Func(Of IVertexQuery, IEnumerable(Of IVertex))
)
```

C++

```
public:
    WrapperVertexQuery(
        IVertexQuery^ query,
        Func<IVertexQuery^, IEnumerable<IEdge^>>^ edgesSelector,
        Func<IVertexQuery^, IEnumerable<IVertex^>>^ verticesSelector
    )
```

F#

```
new :
    query : IVertexQuery *
    edgesSelector : Func<IVertexQuery, IEnumerable<IEdge>> *
    verticesSelector : Func<IVertexQuery, IEnumerable<IVertex>> ->
WrapperVertexQuery
```

Parameters

query

Type: [VelocityGraph.Frontenac.Blueprints.IVertexQuery](#)

[Missing <param name="query"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

edgesSelector

Type: [System.Func\(IVertexQuery, IEnumerable\(IEdge\)\)](#)

[Missing <param name="edgesSelector"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

verticesSelector

Type: [System.Func\(IVertexQuery, IEnumerable\(IVertex\)\)](#)

[Missing <param name="verticesSelector"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.#ctor(VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge}},System.Func{VelocityGraph.Frontenac.Blueprints.IVertexQuery,System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex}})"]

See Also











[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.WrappperVertexQuery Methods

The [WrapperVertexQuery](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Count | |
|  | Direction | |
|  | Edges | |
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |
|  | Interval(T) | |
|  | Labels | |
|  | Limit | |
|  | VertexIds | |
|  | Vertices | |

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Count Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Count"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count ()
```

VB

```
Public Function Count As Long
```

C++

```
public:  
virtual long long Count () sealed
```

F#

```
abstract Count : unit -> int64  
override Count : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Count"]

Implements

[IVertexQuery.Count\(\)](#)

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Direction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Direction(  
    Direction direction  
)
```

VB

```
Public Function Direction (  
    direction As Direction  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Direction(  
    Direction direction  
) sealed
```

F#

```
abstract Direction :  
    direction : Direction -> IVertexQuery  
override Direction :  
    direction : Direction -> IVertexQuery
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Direction(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IVertexQuery.Direction\(Direction\)](#)

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Edges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Edges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> Edges ()
```

VB

```
Public Function Edges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ Edges () sealed
```

F#

```
abstract Edges : unit -> IEnumerable<IEdge>  
override Edges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Edges"]

Implements

[IQuery.Edges\(\)](#)

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Has Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------|-------------|
|  | Has(String, Object) | |
|  | Has(T)(String, Compare, T) | |

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Has Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Has(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Has(  
    string key,  
    Object value  
)
```

VB

```
Public Function Has (  
    key As String,  
    value As Object  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Has(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Has :  
    key : string *  
    value : Object -> IQuery  
override Has :  
    key : string *  
    value : Object -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Has(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Has(System.String,System.Object)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Has(System.String,System.Object)"]

Implements

[IQuery.Has\(String, Object\)](#)

See Also

[WrapperVertexQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Has(T) Method (String, Compare, T)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Has<T>(
    string key,
    Compare compare,
    T value
)
```

VB

```
Public Function Has(Of T) (
    key As String,
    compare As Compare,
    value As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Has(
    String^ key,
    Compare compare,
    T value
) sealed
```

F#

```
abstract Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
override Has :
    key : string *
    compare : Compare *
    value : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

compare

Type: [VelocityGraph.Frontenac.Blueprints.Compare](#)

[Missing <param name="compare"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

value

Type: *T*

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Has`1(System.String,VelocityGraph.Frontenac.Blueprints.Compare,`0)"]

Implements

[IQuery.Has\(T\)\(String, Compare, T\)](#)

See Also

[WrapperVertexQuery Class](#)

[Has Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Interval(T) Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Interval<T>(
    string key,
    T startValue,
    T endValue
)
```

VB

```
Public Function Interval(Of T) (
    key As String,
    startValue As T,
    endValue As T
) As IQuery
```

C++

```
public:
generic<typename T>
virtual IQuery^ Interval(
    String^ key,
    T startValue,
    T endValue
) sealed
```

F#

```
abstract Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
override Interval :
    key : string *
    startValue : 'T *
    endValue : 'T -> IQuery
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

startValue

Type: **T**

[Missing <param name="startValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

endValue

Type: **T**

[Missing <param name="endValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

Type Parameters

T

[Missing <typeparam name="T"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers WrapperVertexQuery.Interval`1(System.String,`0,`0)"]

Implements

[IQuery.Interval\(T\)\(String, T, T\)](#)

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Labels Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Labels(System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Labels(  
    params string[] labels  
)
```

VB

```
Public Function Labels (  
    ParamArray labels As String()  
) As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Labels(  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract Labels :  
    labels : string[] -> IVertexQuery  
override Labels :  
    labels : string[] -> IVertexQuery
```

Parameters

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Labels(System.String[])"]

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Labels(System.String[])"]

Implements

[IVertexQuery.Labels\(String\[\]\)](#)

VelocityDB Class Library

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Limit Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Limit(System.Int64)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Limit(  
    long max  
)
```

VB

```
Public Function Limit (  
    max As Long  
) As IQuery
```

C++

```
public:  
virtual IQuery^ Limit(  
    long long max  
) sealed
```

F#

```
abstract Limit :  
    max : int64 -> IQuery  
override Limit :  
    max : int64 -> IQuery
```

Parameters

max

Type: [System.Int64](#)

[Missing <param name="max"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Limit(System.Int64)"]

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Limit(System.Int64)"]

Implements

[IQuery.Limit\(Int64\)](#)

VelocityDB Class Library

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.VertexIds Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.VertexIds"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<Object> VertexIds ()
```

VB

```
Public Function VertexIds As IEnumerable(Of Object)
```

C++

```
public:  
virtual IEnumerable<Object^>^ VertexIds () sealed
```

F#

```
abstract VertexIds : unit -> IEnumerable<Object>  
override VertexIds : unit -> IEnumerable<Object>
```

Return Value

Type: [IEnumerable\(Object\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.VertexIds"]

Implements

[IVertexQuery.VertexIds\(\)](#)

See Also

[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)

WrapperVertexQuery.Vertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Vertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> Vertices ()
```

VB

```
Public Function Vertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ Vertices () sealed
```

F#

```
abstract Vertices : unit -> IEnumerable<IVertex>  
override Vertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.WrapperVertexQuery.Vertices"]

Implements

[IQuery.Vertices\(\)](#)

See Also




[WrapperVertexQuery Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers Namespace](#)


VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace

[Missing <summary> documentation for
"N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch"]

Classes

| Class | Description |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  BatchGraph | <p>BatchGraph is a wrapper that enables batch loading of a large number of edges and vertices by chunking the entire load into smaller batches and maintaining a memory-efficient vertex cache so that the entire transactional state can be flushed after each chunk is loaded.</p> <p>BatchGraph is ONLY meant for loading data and does not support any retrieval or removal operations. That is, BatchGraph only supports the following methods: - addVertex for adding vertices - addEdge for adding edges - getVertex to be used when adding edges - Property getter, setter and removal methods for vertices and edges.</p> <p>An important limitation of BatchGraph is that edge properties can only be set immediately after the edge has been added. If other vertices or edges have been created in the meantime, setting, getting or removing properties will throw exceptions. This is done to avoid caching of edges which would require a great amount of memory.</p> <p>BatchGraph wraps TransactionalGraph. To wrap arbitrary graphs, use wrap which will additionally wrap non-transactional.</p> <p>BatchGraph can also automatically set the provided element ids as properties on the respective element. Use setVertexIdKey and setEdgeIdKey to set the keys for the vertex and edge properties respectively. This allows to make the loaded baseGraph compatible for later wrapping with idInnerTinkerGraph.</p> |
|  VertexIdTypes | |
|  WritethroughGraph | <p>This is a naive wrapper to make a non-transactional graph transactional by simply writing all mutations directly through to the wrapped graph and not supporting transactional failures.</p> <p>Hence, this is not meant as a functional implementation of a TransactionalGraph but rather as a means to using non-transactional graphs where transactional graphs are expected and transactional failure can be excluded. BatchGraph is one such case.</p> <p>Note, the constructor will throw an exception if the given graph already supports transactions.</p> |

Enumerations

| Enumeration | Description |
|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  VertexIdType | Type of vertex ids expected by BatchGraph. The default is IdType.OBJECT. Use the IdType that best matches the used vertex id types in order to save memory. |

BatchGraph Class

BatchGraph is a wrapper that enables batch loading of a large number of edges and vertices by chunking the entire load into smaller batches and maintaining a memory-efficient vertex cache so that the entire transactional state can be flushed after each chunk is loaded.

BatchGraph is ONLY meant for loading data and does not support any retrieval or removal operations. That is, BatchGraph only supports the following methods: - addVertex for adding vertices - addEdge for adding edges - getVertex to be used when adding edges - Property getter, setter and removal methods for vertices and edges.

An important limitation of BatchGraph is that edge properties can only be set immediately after the edge has been added. If other vertices or edges have been created in the meantime, setting, getting or removing properties will throw exceptions. This is done to avoid caching of edges which would require a great amount of memory.

BatchGraph wraps TransactionalGraph. To wrap arbitrary graphs, use wrap which will additionally wrap non-transactional.

BatchGraph can also automatically set the provided element ids as properties on the respective element. Use setVertexIdKey and setEdgeIdKey to set the keys for the vertex and edge properties respectively. This allows to make the loaded baseGraph compatible for later wrapping with idInnerTinkerGraph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class BatchGraph : ITransactionalGraph, IGraph,
    IWrapperGraph
```

VB

```
Public Class BatchGraph
    Implements ITransactionalGraph, IGraph, IWrapperGraph
```

C++

```
public ref class BatchGraph : ITransactionalGraph,
    IGraph, IWrapperGraph
```

F#

```
type BatchGraph =
    class
        interface ITransactionalGraph
        interface IGraph
```



```

interface IWrapperGraph
end


```

The **BatchGraph** type exposes the following members.













Constructors














| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  BatchGraph(ITransactionalGraph) | Constructs a BatchGraph wrapping the provided baseGraph. |
|  BatchGraph(ITransactionalGraph, VertexIdType, Int64) | Constructs a BatchGraph wrapping the provided baseGraph, using the specified buffer size and expecting vertex ids of the specified IdType. Supplying vertex ids which do not match this type will throw exceptions. |

Properties

| Name | Description |
|------------------------------------------------------------------------------------------------------------|-------------|
|  Features | |

Methods


| Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | |
|  AddVertex | |
|  Commit | Should only be invoked after loading is complete. Committing the transaction before will cause the loading to fail. |
|  GetBaseGraph | |
|  GetEdge | |
|  GetEdgeIdKey | Returns the key used to set the id on the edges or null if such has not been set via setEdgeIdKey |
|  GetEdges() | |
|  GetEdges(String, Object) | |
|  GetVertex | |
|  GetVertexIdKey | Returns the key used to set the id on the vertices or null if such has not been set via setVertexIdKey |
|  GetVertices() | |
|  GetVertices(String, Object) | |
|  IsLoadingFromScratch | Whether this BatchGraph is loading data from scratch or incrementally into an existing Graph. By default, this returns true. see setLoadingFromScratch |













| | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | Not supported for batch loading, since data may have already been partially persisted. |
|  | SetEdgeIdKey | Sets the key to be used when setting the edge id as a property on the respective edge. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with IdGraph use idInnerTinkerGraph.ID. |
|  | SetLoadingFromScratch | Sets whether the Graph loaded through this instance of BatchGraph is loaded from scratch (i.e. the wrapped Graph is initially empty) or whether Graph is loaded incrementally into an existing Graph. In the former case, BatchGraph does not need to check for the existence of vertices with the wrapped Graph but only needs to consult its own cache which can be significantly faster. In the latter case, the cache is checked first but an additional check against the wrapped Graph may be necessary if the vertex does not exist. By default, BatchGraph assumes that the data is loaded from scratch. When setting loading from scratch to false, a vertex id key must be specified first using setVertexIdKey - otherwise an exception is thrown. |
|  | SetVertexIdKey | Sets the key to be used when setting the vertex id as a property on the respective vertex. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with idInnerTinkerGraph use idInnerTinkerGraph.ID. |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |
|   | Wrap(IGraph) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |
|   | Wrap(IGraph, Int64) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|---------------------|
|  | DefaultBufferSize | Default buffer size |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |



| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | BatchGraph(ITransactionalGraph) | Constructs a BatchGraph wrapping the provided baseGraph. |
|  | BatchGraph(ITransactionalGraph, VertexIdType, Int64) | Constructs a BatchGraph wrapping the provided baseGraph, using the specified buffer size and expecting vertex ids of the specified IdType. Supplying vertex ids which do not match this type will throw exceptions. |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph Constructor (ITransactionalGraph)

Constructs a BatchGraph wrapping the provided baseGraph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BatchGraph(  
    ITransactionalGraph graph  
)
```

VB

```
Public Sub New (  
    graph As ITransactionalGraph  
)
```

C++

```
public:  
BatchGraph(  
    ITransactionalGraph^ graph  
)
```

F#

```
new :  
    graph : ITransactionalGraph -> BatchGraph
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.ITransactionalGraph](#)

Graph to be wrapped

See Also

[BatchGraph Class](#)

[BatchGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph Constructor (ITransactionalGraph, VertexIdType, Int64)

Constructs a BatchGraph wrapping the provided baseGraph, using the specified buffer size and expecting vertex ids of the specified IdType. Supplying vertex ids which do not match this type will throw exceptions.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public BatchGraph(  
    ITransactionalGraph graph,  
    VertexIdType type,  
    long bufferSize  
)
```

VB

```
Public Sub New (  
    graph As ITransactionalGraph,  
    type As VertexIdType,  
    bufferSize As Long  
)
```

C++

```
public:  
BatchGraph(  
    ITransactionalGraph^ graph,  
    VertexIdType type,  
    long long bufferSize  
)
```

F#

```
new :  
    graph : ITransactionalGraph *  
    type : VertexIdType *  
    bufferSize : int64 -> BatchGraph
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.ITransactionalGraph](#)

Graph to be wrapped

type

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdType](#)

Type of vertex id expected. This information is used to optimize the vertex cache memory footprint.

bufferSize

Type: [System.Int64](#)

Defines the number of vertices and edges loaded before starting a new transaction. The larger this value, the more memory is required but the faster the loading process.

See Also

[BatchGraph Class](#)


[BatchGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.BatchGraph Properties

The [BatchGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[BatchGraph Class](#)








[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.BatchGraph Methods










The [BatchGraph](#) type exposes the following members.

Methods

| Name | Description |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AddEdge | |
| AddVertex | |
| Commit | Should only be invoked after loading is complete. Committing the transaction before will cause the loading to fail. |
| GetBaseGraph | |
| GetEdge | |
| GetEdgeldKey | Returns the key used to set the id on the edges or null if such has not been set via setEdgeldKey |
| GetEdges() | |
| GetEdges(String, Object) | |
| GetVertex | |
| GetVertexIdKey | Returns the key used to set the id on the vertices or null if such has not been set via setVertexIdKey |
| GetVertices() | |
| GetVertices(String, Object) | |
| IsLoadingFromScratch | Whether this BatchGraph is loading data from scratch or incrementally into an existing Graph. By default, this returns true. see setLoadingFromScratch |
| Query | |
| RemoveEdge | |
| RemoveVertex | |
| Rollback | Not supported for batch loading, since data may have already been partially persisted. |
| SetEdgeldKey | Sets the key to be used when setting the edge id as a property on the respective edge. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with IdGraphuse idInnerTinkerGraph.ID. |
| SetLoadingFromScratch | Sets whether the Graph loaded through this instance of BatchGraph is loaded from scratch (i.e. the wrapped Graph is initially empty) or whether Graph is loaded incrementally into an existing Graph. In the former case, BatchGraph does not need to check for the existence of vertices with the wrapped Graph but only needs to consult its own cache which can be significantly faster. In the latter case, the cache is checked first but an |

| | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | additional check against the wrapped Graph may be necessary if the vertex does not exist. By default, BatchGraph assumes that the data is loaded from scratch. When setting loading from scratch to false, a vertex id key must be specified first using <code>setVertexIdKey</code> - otherwise an exception is thrown. |
|  | SetVertexIdKey | Sets the key to be used when setting the vertex id as a property on the respective vertex. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with <code>idInnerTinkerGraph</code> use <code>idInnerTinkerGraph.ID</code> . |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |
|   | Wrap(IGraph) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |
|   | Wrap(IGraph, Int64) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |

VelocityDB Class Library

| | |
|----------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.AddVertex(System.Object)"]

Implements

[IGraph.AddVertex\(Object\)](#)

VelocityDB Class Library

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Commit Method

Should only be invoked after loading is complete. Committing the transaction before will cause the loading to fail.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
virtual void Commit () sealed
```

F#

```
abstract Commit : unit -> unit  
override Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (  
    Object id  
)
```

VB

```
Public Function GetEdge (  
    id As Object  
) As IEdge
```

C++

```
public:  
virtual IEdge^ GetEdge (  
    Object^ id  
) sealed
```

F#

```
abstract GetEdge :  
    id : Object -> IEdge  
override GetEdge :  
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdge(System.Object)"]

Implements

[IGraph.GetEdge\(Object\)](#)

VelocityDB Class Library

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetEdgeIdKey Method

Returns the key used to set the id on the edges or null if such has not been set via setEdgeIdKey

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string GetEdgeIdKey()
```

VB

```
Public Function GetEdgeIdKey As String
```

C++

```
public:  
String^ GetEdgeIdKey()
```

F#

```
member GetEdgeIdKey : unit -> string
```

Return Value

Type: [String](#)

The key used to set the id on the edges or null if such has not been set

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetEdges Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[BatchGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[BatchGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertex(System.Object)"]

Implements

[IGraph.GetVertex\(Object\)](#)

VelocityDB Class Library

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetVertexIdKey Method

Returns the key used to set the id on the vertices or null if such has not been set via setVertexIdKey

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string GetVertexIdKey()
```

VB

```
Public Function GetVertexIdKey As String
```

C++

```
public:  
String^ GetVertexIdKey()
```

F#

```
member GetVertexIdKey : unit -> string
```

Return Value

Type: [String](#)

The key used to set the id on the vertices or null if such has not been set

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[BatchGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
override GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[BatchGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.IsLoadingFromScratch Method

Whether this BatchGraph is loading data from scratch or incrementally into an existing Graph. By default, this returns true. see setLoadingFromScratch

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsLoadingFromScratch ()
```

VB

```
Public Function IsLoadingFromScratch As Boolean
```

C++

```
public:  
bool IsLoadingFromScratch ()
```

F#

```
member IsLoadingFromScratch : unit -> bool
```

Return Value

Type: [Boolean](#)

Whether this BatchGraph is loading data from scratch or incrementally into an existing Graph.

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Public Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void RemoveEdge (  
    IEdge^ edge  
) sealed
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit  
override RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Rollback Method

Not supported for batch loading, since data may have already been partially persisted.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
virtual void Rollback() sealed
```

F#

```
abstract Rollback : unit -> unit  
override Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.SetEdgeIdKey Method

Sets the key to be used when setting the edge id as a property on the respective edge. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with IdGraphuse idInnerTinkerGraph.ID.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetEdgeIdKey (
    string key
)
```

VB

```
Public Sub SetEdgeIdKey (
    key As String
)
```

C++

```
public:
void SetEdgeIdKey (
    String^ key
)
```

F#

```
member SetEdgeIdKey :
    key : string -> unit
```

Parameters

key

Type: [System.String](#)

Key to be used.

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.SetLoadingFromScratch Method

Sets whether the Graph loaded through this instance of BatchGraph is loaded from scratch (i.e. the wrapped Graph is initially empty) or whether Graph is loaded incrementally into an existing Graph. In the former case, BatchGraph does not need to check for the existence of vertices with the wrapped Graph but only needs to consult its own cache which can be significantly faster. In the latter case, the cache is checked first but an additional check against the wrapped Graph may be necessary if the vertex does not exist. By default, BatchGraph assumes that the data is loaded from scratch. When setting loading from scratch to false, a vertex id key must be specified first using `setVertexIdKey` - otherwise an exception is thrown.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetLoadingFromScratch (  
    bool fromScratch  
)
```

VB

```
Public Sub SetLoadingFromScratch (  
    fromScratch As Boolean  
)
```

C++

```
public:  
void SetLoadingFromScratch (  
    bool fromScratch  
)
```

F#

```
member SetLoadingFromScratch :  
    fromScratch : bool -> unit
```

Parameters

fromScratch

Type: [System.Boolean](#)

Sets whether the Graph loaded through this instance of BatchGraph is loaded from scratch

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.SetVertexIdKey Method

Sets the key to be used when setting the vertex id as a property on the respective vertex. If the key is null, then no property will be set. If the loaded baseGraph should later be wrapped with idInnerTinkerGraph use idInnerTinkerGraph.ID.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetVertexIdKey (
    string key
)
```

VB

```
Public Sub SetVertexIdKey (
    key As String
)
```

C++

```
public:
void SetVertexIdKey (
    String^ key
)
```

F#

```
member SetVertexIdKey :
    key : string -> unit
```

Parameters

key

Type: [System.String](#)

key Key to be used.

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.ToString Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.BatchGraph.ToString"]





See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Wrap Method

Overload List

| | Name | Description |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | Wrap(IGraph) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |
|   | Wrap(IGraph, Int64) | Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph. |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Wrap Method (IGraph)

Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static BatchGraph Wrap(  
    IGraph graph  
)
```

VB

```
Public Shared Function Wrap (  
    graph As IGraph  
) As BatchGraph
```

C++

```
public:  
static BatchGraph^ Wrap(  
    IGraph^ graph  
)
```

F#

```
static member Wrap :  
    graph : IGraph -> BatchGraph
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

Graph to be wrapped

Return Value

Type: [BatchGraph](#)

a BatchGraph wrapping the provided baseGraph

See Also

[BatchGraph Class](#)

[Wrap Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.Wrap Method (IGraph, Int64)

Constructs a BatchGraph wrapping the provided baseGraph. Immediately returns the baseGraph if its a BatchGraph and wraps non-transactional graphs in an additional WritethroughGraph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static BatchGraph Wrap(  
    IGraph graph,  
    long buffer  
)
```

VB

```
Public Shared Function Wrap (  
    graph As IGraph,  
    buffer As Long  
) As BatchGraph
```

C++

```
public:  
static BatchGraph^ Wrap(  
    IGraph^ graph,  
    long long buffer  
)
```

F#

```
static member Wrap :  
    graph : IGraph *  
    buffer : int64 -> BatchGraph
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

Graph to be wrapped

buffer

Type: [System.Int64](#)

Size of the buffer

Return Value

Type: [BatchGraph](#)

a BatchGraph wrapping the provided baseGraph

VelocityDB Class Library

See Also

[BatchGraph Class](#)

[Wrap Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.BatchGraph Fields

The [BatchGraph](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|---------------------|
|  | DefaultBufferSize | Default buffer size |

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

BatchGraph.DefaultBufferSize Field

Default buffer size

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const long DefaultBufferSize = 100000
```

VB

```
Public Const DefaultBufferSize As Long = 100000
```

C++

```
public:  
literal long long DefaultBufferSize = 100000
```

F#

```
static val mutable DefaultBufferSize: int64
```

Field Value

Type: [Int64](#)

See Also

[BatchGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

VertexIdType Enumeration

Type of vertex ids expected by BatchGraph. The default is IdType.OBJECT. Use the IdType that best matches the used vertex id types in order to save memory.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public enum VertexIdType
```

VB

```
Public Enumeration VertexIdType
```

C++

```
public enum class VertexIdType
```

F#

```
type VertexIdType
```

Members

| Member name | Value | Description |
|---------------|-------|-------------|
| Object | 0 | |
| Number | 1 | |
| String | 2 | |
| Url | 3 | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

VertexIdTypes Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdTypes](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdTypes

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class VertexIdTypes
```

VB

```
<ExtensionAttribute>  
Public NotInheritable Class VertexIdTypes
```

C++

```
[ExtensionAttribute]  
public ref class VertexIdTypes abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
[<ExtensionAttribute>]  
type VertexIdTypes = class end
```

The **VertexIdTypes** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | GetVertexCache | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

VertexIdTypes.VertexIdTypes Methods

The [VertexIdTypes](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | GetVertexCache | |

See Also

[VertexIdTypes Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

VertexIdTypes.GetVertexCache Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdTypes.GetVertexCache(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdType)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static IVertexCache GetVertexCache (  
    this VertexIdType vertexIdType  
)
```

VB

```
<ExtensionAttribute>  
Public Shared Function GetVertexCache (  
    vertexIdType As VertexIdType  
) As IVertexCache
```

C++

```
public:  
[ExtensionAttribute]  
static IVertexCache^ GetVertexCache (  
    VertexIdType vertexIdType  
)
```

F#

```
[<ExtensionAttribute>]  
static member GetVertexCache :  
    vertexIdType : VertexIdType -> IVertexCache
```

Parameters

vertexIdType

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdType](#)

[Missing <param name="vertexIdType"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdTypes.GetVertexCache(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdType)"]

Return Value

Type: [IVertexCache](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdTypes.GetVertexCache(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.VertexIdType)"]

VelocityDB Class Library

Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type [VertexIdType](#). When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods \(Visual Basic\)](#) or [Extension Methods \(C# Programming Guide\)](#).

See Also

[VertexIdTypes Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph Class

This is a naive wrapper to make a non-transactional graph transactional by simply writing all mutations directly through to the wrapped graph and not supporting transactional failures.

Hence, this is not meant as a functional implementation of a TransactionalGraph but rather as a means to using non-transactional graphs where transactional graphs are expected and transactional failure can be excluded. BatchGraph is one such case.

Note, the constructor will throw an exception if the given graph already supports transactions.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WritethroughGraph : IWrapperGraph,
    ITransactionalGraph, IGraph
```

VB

```
Public Class WritethroughGraph
    Implements IWrapperGraph, ITransactionalGraph, IGraph
```

C++


```
public ref class WritethroughGraph : IWrapperGraph,
    ITransactionalGraph, IGraph
```

F#


```
type WritethroughGraph =
    class
        interface IWrapperGraph
        interface ITransactionalGraph
        interface IGraph
    end
```

The **WritethroughGraph** type exposes the following members.








Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | WritethroughGraph | Initializes a new instance of the WritethroughGraph class |







Properties








| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------|
|  | Features | Returns the features of the underlying graph but with transactions supported. |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph Constructor

Initializes a new instance of the [WritethroughGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WritethroughGraph(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
WritethroughGraph(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> WritethroughGraph
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also


[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.WritethroughGraph Properties

The [WritethroughGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------------------------------------|
|  | Features | Returns the features of the underlying graph but with transactions supported. |

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.Features Property

Returns the features of the underlying graph but with transactions supported.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

The features of the underlying graph but with transactions supported

Implements

[IGraph.Features](#)

See Also












[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)









WritethroughGraph.WritethroughGraph Methods






The [WritethroughGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.AddVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.AddVertex\(Object\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.Commit Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.Commit"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
virtual void Commit () sealed
```

F#

```
abstract Commit : unit -> unit  
override Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdge(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetEdge\(Object\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[WritethroughGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    string key,
    Object value
)
```

VB

```
Public Function GetEdges (
    key As String,
    value As Object
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
override GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[WritethroughGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetVertex\(Object\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[WritethroughGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
override GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[WritethroughGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (
    IEdge edge
)
```

VB

```
Public Sub RemoveEdge (
    edge As IEdge
)
```

C++

```
public:
virtual void RemoveEdge (
    IEdge^ edge
) sealed
```

F#

```
abstract RemoveEdge :
    edge : IEdge -> unit
override RemoveEdge :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.Rollback Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.Rollback"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
virtual void Rollback() sealed
```

F#

```
abstract Rollback : unit -> unit  
override Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Shutdown ()
```

VB

```
Public Overridable Sub Shutdown
```

C++

```
public:  
virtual void Shutdown ()
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

WritethroughGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.WritethroughGraph.ToString"]

See Also








[WritethroughGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace

[Missing <summary> documentation for
"N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-------------------------------------------|-------------|
|  | LongIdVertexCache | |
|  | ObjectIdVertexCache | |
|  | StringCompression | |
|  | StringCompressionContract | |
|  | StringIdVertexCache | |
|  | UrlCompression | |
|  | VertexCacheContract | |

Interfaces

| | Interface | Description |
|------------------------------------------------------------------------------------|------------------------------|-------------|
|  | IVertexCache | |

IVertexCache Interface

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#  
public interface IVertexCache
```






```
VB  
Public Interface IVertexCache
```

```
C++  
public interface class IVertexCache
```

```
F#  
type IVertexCache = interface end
```

The **IVertexCache** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |






See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.IVertexCache Methods

The [IVertexCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.Contains Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Contains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
bool Contains(  
    Object externalId  
)
```

VB

```
Function Contains (  
    externalId As Object  
) As Boolean
```

C++

```
bool Contains(  
    Object^ externalId  
)
```

F#

```
abstract Contains :  
    externalId : Object -> bool
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Contains(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Contains(System.Object)"]

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.GetEntry Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.GetEntry(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object GetEntry(  
    Object externalId  
)
```

VB

```
Function GetEntry (  
    externalId As Object  
) As Object
```

C++

```
Object^ GetEntry(  
    Object^ externalId  
)
```

F#

```
abstract GetEntry :  
    externalId : Object -> Object
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.GetEntry(System.Object)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.GetEntry(System.Object)"]

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.NewTransaction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.NewTransaction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void NewTransaction()
```

VB

```
Sub NewTransaction
```

C++

```
void NewTransaction()
```

F#

```
abstract NewTransaction : unit -> unit
```

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.Set Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void Set (  
    IVertex vertex,  
    Object externalId  
)
```

VB

```
Sub Set (  
    vertex As IVertex,  
    externalId As Object  
)
```

C++

```
void Set (  
    IVertex^ vertex,  
    Object^ externalId  
)
```

F#

```
abstract Set :  
    vertex : IVertex *  
    externalId : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

VelocityDB Class Library

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

IVertexCache.SetId Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.SetId(System.Object ,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void SetId(  
    Object vertexId,  
    Object externalId  
)
```

VB

```
Sub SetId (  
    vertexId As Object,  
    externalId As Object  
)
```

C++

```
void SetId(  
    Object^ vertexId,  
    Object^ externalId  
)
```

F#

```
abstract SetId :  
    vertexId : Object *  
    externalId : Object -> unit
```

Parameters

vertexId

Type: [System.Object](#)

[Missing <param name="vertexId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.SetId(System.Object ,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.IVertexCache.SetId(System.Object ,System.Object)"]

VelocityDB Class Library

See Also

[IVertexCache Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class LongIdVertexCache : IVertexCache
```


```
VB
Public Class LongIdVertexCache
    Implements IVertexCache
```

```
C++
public ref class LongIdVertexCache : IVertexCache
```






```
F#
type LongIdVertexCache =
    class
        interface IVertexCache
    end
```

The **LongIdVertexCache** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | LongIdVertexCache | Initializes a new instance of the LongIdVertexCache class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache Constructor

Initializes a new instance of the [LongIdVertexCache](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public LongIdVertexCache ()
```

VB

```
Public Sub New
```

C++

```
public:  
LongIdVertexCache ()
```

F#

```
new : unit -> LongIdVertexCache
```

See Also






[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.LongIdVertexCache Methods

The [LongIdVertexCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.Contains Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Contains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Object externalId  
)
```

VB

```
Public Function Contains (  
    externalId As Object  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Object^ externalId  
) sealed
```

F#

```
abstract Contains :  
    externalId : Object -> bool  
override Contains :  
    externalId : Object -> bool
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Contains(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Contains(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.Contains\(Object\)](#)

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.GetEntry Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.GetEntry(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetEntry(  
    Object externalId  
)
```

VB

```
Public Function GetEntry (  
    externalId As Object  
) As Object
```

C++

```
public:  
virtual Object^ GetEntry(  
    Object^ externalId  
) sealed
```

F#

```
abstract GetEntry :  
    externalId : Object -> Object  
override GetEntry :  
    externalId : Object -> Object
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.GetEntry(System.Object)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.GetEntry(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.GetEntry\(Object\)](#)

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.NewTransaction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.NewTransaction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void NewTransaction()
```

VB

```
Public Sub NewTransaction
```

C++

```
public:  
virtual void NewTransaction() sealed
```

F#

```
abstract NewTransaction : unit -> unit  
override NewTransaction : unit -> unit
```

Implements

[IVertexCache.NewTransaction\(\)](#)

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.Set Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Set (
    IVertex vertex,
    Object externalId
)
```

VB

```
Public Sub Set (
    vertex As IVertex,
    externalId As Object
)
```

C++

```
public:
virtual void Set (
    IVertex^ vertex,
    Object^ externalId
) sealed
```

F#

```
abstract Set :
    vertex : IVertex *
    externalId : Object -> unit
override Set :
    vertex : IVertex *
    externalId : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Implements

[IVertexCache.Set\(IVertex, Object\)](#)

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

LongIdVertexCache.SetId Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.SetId(System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetId(  
    Object vertexId,  
    Object externalId  
)
```

VB

```
Public Sub SetId (  
    vertexId As Object,  
    externalId As Object  
)
```

C++

```
public:  
virtual void SetId(  
    Object^ vertexId,  
    Object^ externalId  
) sealed
```

F#

```
abstract SetId :  
    vertexId : Object *  
    externalId : Object -> unit  
override SetId :  
    vertexId : Object *  
    externalId : Object -> unit
```

Parameters

vertexId

Type: [System.Object](#)

[Missing <param name="vertexId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.SetId(System.Object,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.LongIdVertexCache.SetId(System.Object,System.Object)"]

Implements

[IVertexCache.SetId\(Object, Object\)](#)

See Also

[LongIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ObjectIdVertexCache : IVertexCache
```

VB

```
Public Class ObjectIdVertexCache
    Implements IVertexCache
```

C++


```
public ref class ObjectIdVertexCache : IVertexCache
```

F#






```
type ObjectIdVertexCache =
    class
        interface IVertexCache
    end
```

The **ObjectIdVertexCache** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------|
|  | ObjectIdVertexCache | Initializes a new instance of the ObjectIdVertexCache class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache Constructor

Initializes a new instance of the [ObjectIdVertexCache](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ObjectIdVertexCache ()
```

VB

```
Public Sub New
```

C++

```
public:  
ObjectIdVertexCache ()
```

F#

```
new : unit -> ObjectIdVertexCache
```

See Also






[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.ObjectIdVertexCache Methods

The [ObjectIdVertexCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.Contains Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Contains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Object externalId  
)
```

VB

```
Public Function Contains (  
    externalId As Object  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Object^ externalId  
) sealed
```

F#

```
abstract Contains :  
    externalId : Object -> bool  
override Contains :  
    externalId : Object -> bool
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Contains(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Contains(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.Contains\(Object\)](#)

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.GetEntry Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.GetEntry(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetEntry(  
    Object externalId  
)
```

VB

```
Public Function GetEntry (  
    externalId As Object  
) As Object
```

C++

```
public:  
virtual Object^ GetEntry(  
    Object^ externalId  
) sealed
```

F#

```
abstract GetEntry :  
    externalId : Object -> Object  
override GetEntry :  
    externalId : Object -> Object
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.GetEntry(System.Object)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.GetEntry(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.GetEntry\(Object\)](#)

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.NewTransaction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.NewTransaction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void NewTransaction()
```

VB

```
Public Sub NewTransaction
```

C++

```
public:  
virtual void NewTransaction() sealed
```

F#

```
abstract NewTransaction : unit -> unit  
override NewTransaction : unit -> unit
```

Implements

[IVertexCache.NewTransaction\(\)](#)

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.Set Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Set (
    IVertex vertex,
    Object externalId
)
```

VB

```
Public Sub Set (
    vertex As IVertex,
    externalId As Object
)
```

C++

```
public:
virtual void Set(
    IVertex^ vertex,
    Object^ externalId
) sealed
```

F#

```
abstract Set :
    vertex : IVertex *
    externalId : Object -> unit
override Set :
    vertex : IVertex *
    externalId : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Implements

[IVertexCache.Set\(IVertex, Object\)](#)

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

ObjectIdVertexCache.SetId Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.SetId(System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetId(  
    Object vertexId,  
    Object externalId  
)
```

VB

```
Public Sub SetId (  
    vertexId As Object,  
    externalId As Object  
)
```

C++

```
public:  
virtual void SetId(  
    Object^ vertexId,  
    Object^ externalId  
) sealed
```

F#

```
abstract SetId :  
    vertexId : Object *  
    externalId : Object -> unit  
override SetId :  
    vertexId : Object *  
    externalId : Object -> unit
```

Parameters

vertexId

Type: [System.Object](#)

[Missing <param name="vertexId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.SetId(System.Object,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.ObjectIdVertexCache.SetId(System.Object,System.Object)"]

Implements

[IVertexCache.SetId\(Object, Object\)](#)

See Also

[ObjectIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompression Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class StringCompression
```

VB

```
Public MustInherit Class StringCompression
```

C++


```
public ref class StringCompression abstract
```

F#

```
[<AbstractClassAttribute>]  
type StringCompression = class end
```

The **StringCompression** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Compress | |

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-------------|
|  | NoCompression | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompression.StringCompression Methods

The [StringCompression](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Compress | |

See Also

[StringCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompression.Compress Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression.Compress(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract string Compress(  
    string input  
)
```

VB

```
Public MustOverride Function Compress (  
    input As String  
) As String
```

C++

```
public:  
virtual String^ Compress(  
    String^ input  
) abstract
```

F#

```
abstract Compress :  
    input : string -> string
```

Parameters

input

Type: [System.String](#)

[Missing <param name="input"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression.Compress(System.String)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression.Compress(System.String)"]

See Also


[StringCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompression.StringCompression Fields

The [StringCompression](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | NoCompression | |

See Also

[StringCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompression.NoCompression Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression.NoCompression"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static StringCompression NoCompression
```

VB

```
Public Shared NoCompression As StringCompression
```

C++

```
public:  
static StringCompression^ NoCompression
```

F#

```
static val mutable NoCompression: StringCompression
```

Field Value

Type: [StringCompression](#)

See Also

[StringCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompressionContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompressionContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompressionContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class StringCompressionContract
```

VB

```
Public NotInheritable Class StringCompressionContract
```

C++

```
public ref class StringCompressionContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type StringCompressionContract = class end
```

The **StringCompressionContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | ValidateCompress | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompressionContract.StringCompressionContract Methods

The [StringCompressionContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-------------|
|  | ValidateCompress | |

See Also

[StringCompressionContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringCompressionContract.ValidateCompress Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompressionContract.ValidateCompress(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateCompress (  
    string input  
)
```

VB

```
Public Shared Sub ValidateCompress (  
    input As String  
)
```

C++

```
public:  
static void ValidateCompress (  
    String^ input  
)
```

F#

```
static member ValidateCompress :  
    input : string -> unit
```

Parameters

input

Type: [System.String](#)

[Missing <param name="input"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompressionContract.ValidateCompress(System.String)"]

See Also

[StringCompressionContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class StringIdVertexCache : IVertexCache
```

VB

```
Public Class StringIdVertexCache
    Implements IVertexCache
```

C++



```
public ref class StringIdVertexCache : IVertexCache
```

F#





```
type StringIdVertexCache =
    class
        interface IVertexCache
    end
```

The **StringIdVertexCache** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------------------|
|  | StringIdVertexCache() | Initializes a new instance of the StringIdVertexCache class |
|  | StringIdVertexCache(StringCompression) | Initializes a new instance of the StringIdVertexCache class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |



| | | |
|-----------------------------------------------------------------------------------|-----------------------|--|
|  | SetId | |
|-----------------------------------------------------------------------------------|-----------------------|--|

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------|
|  | StringIdVertexCache() | Initializes a new instance of the StringIdVertexCache class |
|  | StringIdVertexCache(StringCompression) | Initializes a new instance of the StringIdVertexCache class |

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache Constructor

Initializes a new instance of the [StringIdVertexCache](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public StringIdVertexCache ()
```

VB

```
Public Sub New
```

C++

```
public:  
StringIdVertexCache ()
```

F#

```
new : unit -> StringIdVertexCache
```

See Also

[StringIdVertexCache Class](#)

[StringIdVertexCache Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache Constructor (StringCompression)

Initializes a new instance of the [StringIdVertexCache](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public StringIdVertexCache(  
    StringCompression compression  
)
```

VB

```
Public Sub New (  
    compression As StringCompression  
)
```

C++

```
public:  
StringIdVertexCache(  
    StringCompression^ compression  
)
```

F#

```
new :  
    compression : StringCompression -> StringIdVertexCache
```

Parameters

compression

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression](#)

[Missing <param name="compression"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression)"]

See Also

[StringIdVertexCache Class](#)






[StringIdVertexCache Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.StringIdVertexCache Methods

The [StringIdVertexCache](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Contains | |
|  | GetEntry | |
|  | NewTransaction | |
|  | Set | |
|  | SetId | |

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.Contains Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Contains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool Contains(  
    Object externalId  
)
```

VB

```
Public Function Contains (  
    externalId As Object  
) As Boolean
```

C++

```
public:  
virtual bool Contains(  
    Object^ externalId  
) sealed
```

F#

```
abstract Contains :  
    externalId : Object -> bool  
override Contains :  
    externalId : Object -> bool
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Contains(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Contains(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.Contains\(Object\)](#)

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.GetEntry Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.GetEntry(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Object GetEntry(  
    Object externalId  
)
```

VB

```
Public Function GetEntry (  
    externalId As Object  
) As Object
```

C++

```
public:  
virtual Object^ GetEntry(  
    Object^ externalId  
) sealed
```

F#

```
abstract GetEntry :  
    externalId : Object -> Object  
override GetEntry :  
    externalId : Object -> Object
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.GetEntry(System.Object)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.GetEntry(System.Object)"]

VelocityDB Class Library

Implements

[IVertexCache.GetEntry\(Object\)](#)

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.NewTransaction Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.NewTransaction"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void NewTransaction()
```

VB

```
Public Sub NewTransaction
```

C++

```
public:  
virtual void NewTransaction() sealed
```

F#

```
abstract NewTransaction : unit -> unit  
override NewTransaction : unit -> unit
```

Implements

[IVertexCache.NewTransaction\(\)](#)

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.Set Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Set (  
    IVertex vertex,  
    Object externalId  
)
```

VB

```
Public Sub Set (  
    vertex As IVertex,  
    externalId As Object  
)
```

C++

```
public:  
virtual void Set (  
    IVertex^ vertex,  
    Object^ externalId  
) sealed
```

F#

```
abstract Set :  
    vertex : IVertex *  
    externalId : Object -> unit  
override Set :  
    vertex : IVertex *  
    externalId : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.Set(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Implements

[IVertexCache.Set\(IVertex, Object\)](#)

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

StringIdVertexCache.SetId Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.SetId(System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetId(  
    Object vertexId,  
    Object externalId  
)
```

VB

```
Public Sub SetId (  
    vertexId As Object,  
    externalId As Object  
)
```

C++

```
public:  
virtual void SetId(  
    Object^ vertexId,  
    Object^ externalId  
) sealed
```

F#

```
abstract SetId :  
    vertexId : Object *  
    externalId : Object -> unit  
override SetId :  
    vertexId : Object *  
    externalId : Object -> unit
```

Parameters

vertexId

Type: [System.Object](#)

[Missing <param name="vertexId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.SetId(System.Object,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringIdVertexCache.SetId(System.Object,System.Object)"]

Implements

[IVertexCache.SetId\(Object, Object\)](#)

See Also

[StringIdVertexCache Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

UrlCompression Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression](#)"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.StringCompression](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class UrlCompression : StringCompression
```

VB

```
Public Class UrlCompression
    Inherits StringCompression
```

C++


```
public ref class UrlCompression : public StringCompression
```

F#

```
type UrlCompression =
    class
        inherit StringCompression
    end
```

The **UrlCompression** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | UrlCompression | Initializes a new instance of the UrlCompression class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------|
|  | Compress | (Overrides StringCompression.Compress(String) .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

UrlCompression Constructor

Initializes a new instance of the [UrlCompression](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public UrlCompression ()
```

VB

```
Public Sub New
```

C++

```
public:  
UrlCompression ()
```

F#

```
new : unit -> UrlCompression
```

See Also

[UrlCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

UrlCompression.UrlCompression Methods

The [UrlCompression](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------|
|  | Compress | (Overrides StringCompression.Compress(String).) |

See Also

[UrlCompression Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

UrlCompression.Compress Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression.Compress(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string Compress(  
    string input  
)
```

VB

```
Public Overrides Function Compress (  
    input As String  
) As String
```

C++

```
public:  
virtual String^ Compress(  
    String^ input  
) override
```

F#

```
abstract Compress :  
    input : string -> string  
override Compress :  
    input : string -> string
```

Parameters

input

Type: [System.String](#)

[Missing <param name="input"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression.Compress(System.String)"]

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.UrlCompression.Compress(System.String)"]

See Also

[UrlCompression Class](#)

VertexCacheContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class VertexCacheContract
```

VB

```
Public MustInherit Class VertexCacheContract
```

C++





```
public ref class VertexCacheContract abstract
```

F#

```
[<AbstractClassAttribute>]  
type VertexCacheContract = class end
```

The **VertexCacheContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-------------|
|  | ValidateContains | |
|  | ValidateGetEntry | |
|  | ValidateSet | |
|  | ValidateSetId | |





See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VertexCacheContract.VertexCacheContract Methods

The [VertexCacheContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------------|-------------|
|  | ValidateContains | |
|  | ValidateGetEntry | |
|  | ValidateSet | |
|  | ValidateSetId | |

See Also

[VertexCacheContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VertexCacheContract.ValidateContains Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateContains(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateContains (  
    Object externalId  
)
```

VB

```
Public Shared Sub ValidateContains (  
    externalId As Object  
)
```

C++

```
public:  
static void ValidateContains (  
    Object^ externalId  
)
```

F#

```
static member ValidateContains :  
    externalId : Object -> unit
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateContains(System.Object)"]

See Also

[VertexCacheContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VertexCacheContract.ValidateGetEntry Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateGetEntry(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateGetEntry(  
    Object externalId  
)
```

VB

```
Public Shared Sub ValidateGetEntry (  
    externalId As Object  
)
```

C++

```
public:  
static void ValidateGetEntry(  
    Object^ externalId  
)
```

F#

```
static member ValidateGetEntry :  
    externalId : Object -> unit
```

Parameters

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateGetEntry(System.Object)"]

See Also

[VertexCacheContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VertexCacheContract.ValidateSet Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSet(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateSet (
    IVertex vertex,
    Object externalId
)
```

VB

```
Public Shared Sub ValidateSet (
    vertex As IVertex,
    externalId As Object
)
```

C++

```
public:
static void ValidateSet(
    IVertex^ vertex,
    Object^ externalId
)
```

F#

```
static member ValidateSet :
    vertex : IVertex *
    externalId : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSet(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSet(VelocityGraph.Frontenac.Blueprints.IVertex,System.Object)"]

VelocityDB Class Library

See Also

[VertexCacheContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VertexCacheContract.ValidateSetId Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSetId(System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateSetId(  
    Object vertexId,  
    Object externalId  
)
```

VB

```
Public Shared Sub ValidateSetId (  
    vertexId As Object,  
    externalId As Object  
)
```

C++

```
public:  
static void ValidateSetId(  
    Object^ vertexId,  
    Object^ externalId  
)
```

F#

```
static member ValidateSetId :  
    vertexId : Object *  
    externalId : Object -> unit
```

Parameters

vertexId

Type: [System.Object](#)

[Missing <param name="vertexId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSetId(System.Object,System.Object)"]

externalId

Type: [System.Object](#)

[Missing <param name="externalId"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache.VertexCacheContract.ValidateSetId(System.Object,System.Object)"]

VelocityDB Class Library

See Also










[VertexCacheContract Class](#)



[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Batch.Cache Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace

[Missing <summary> documentation for
"N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event"]

Classes

| Class | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  EventEdge | An edge with a GraphChangeListener attached. Those listeners are notified when changes occur to the properties of the edge. |
|  EventEdgelterable | A sequence of edges that applies the list of listeners into each edge. |
|  EventElement | An element with a GraphChangeListener attached. Those listeners are notified when changes occur to the properties of the element. |
|  EventGraph | An eventInnerTinkerGraph is a wrapper to existing Graph implementations and provides for graph events to be raised to one or more listeners on changes to the Graph. Notifications to the listeners occur for the following events: new vertex/edge, vertex/edge property changed, vertex/edge property removed, vertex/edge removed. The limiting factor to events being raised is related to out-of-process functions changing graph elements. To gather events from eventInnerTinkerGraph, simply provide an implementation of the {@link GraphChangeListener} to the eventInnerTinkerGraph by utilizing the addListener method. eventInnerTinkerGraph allows the addition of multiple GraphChangeListener implementations. Each listener will be notified in the order that it was added. |
|  EventIndex | An index that wraps graph elements in the "evented" way. This class does not directly raise graph events, but passes the GraphChangeListener to the edges and vertices returned from indices so that they may raise graph events. |
|  EventIndexableGraph | EventIndexableGraph is merely a proxy to index methods exposing eventInnerTinkerGraph methods in the "evented" way. Like the eventInnerTinkerGraph it extends from, this graph implementations raise notifications to the listeners for the following events: new vertex/edge, vertex/edge property changed, vertex/edge property removed, vertex/edge removed. |
|  EventTransactionalGraph | |
|  EventTransactionalIndexableGraph | The transactional and indexable implementation of eventInnerTinkerGraph where events are raised in batch in the order they changes occurred to the graph, but only after a successful commit to the underlying graph. |
|  EventTrigger | |

| | | |
|-----------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
|  | EventVertex | An vertex with a GraphChangedListener attached. Those listeners are notified when changes occur to the properties of the vertex. |
|  | EventVertexIterable | A sequence of vertices that applies the list of listeners into each vertex. |

EventEdge Class

An edge with a GraphChangedListener attached. Those listeners are notified when changes occur to the properties of the edge.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventEdge : EventElement, IEdge,
    IElement, IDictionary<string, Object>, ICollection<KeyValuePair<string,
Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection
```

VB

```
Public Class EventEdge
    Inherits EventElement
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class EventEdge : public EventElement,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type EventEdge =
    class
        inherit EventElement
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end


```

The **EventEdge** type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|----------------------------------------------------------|
|  | EventEdge | Initializes a new instance of the EventEdge class |








Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | EdgeString | (Defined by StringFactory.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdge Constructor

Initializes a new instance of the [EventEdge](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventEdge (
    IEdge edge,
    EventGraph eventInnerTinkerGraph
)
```

VB

```
Public Sub New (
    edge As IEdge,
    eventInnerTinkerGraph As EventGraph
)
```

C++

```
public:
EventEdge (
    IEdge^ edge,
    EventGraph^ eventInnerTinkerGraph
)
```

F#

```
new :
    edge : IEdge *
    eventInnerTinkerGraph : EventGraph -> EventEdge
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

eventInnerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="eventInnerTinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

See Also

[EventEdge Class](#)


VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdge.EventEdge Properties

The [EventEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

See Also

[EventEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdge.Label Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Label { get; }
```

VB

```
Public ReadOnly Property Label As String  
    Get
```

C++

```
public:  
virtual property String^ Label {  
    String^ get () sealed;  
}
```

F#

```
abstract Label : string with get  
override Label : string with get
```

Property Value

Type: [String](#)

Implements

[IEdge.Label](#)

See Also

[EventEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)











EventEdge.EventEdge Methods

The [EventEdge](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

See Also

[EventEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdge.GetBaseEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.GetBaseEdge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetBaseEdge ()
```

VB

```
Public Function GetBaseEdge As IEdge
```

C++

```
public:  
IEdge^ GetBaseEdge ()
```

F#

```
member GetBaseEdge : unit -> IEdge
```

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.GetBaseEdge"]

See Also

[EventEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdge.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Public Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Direction direction  
) sealed
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex  
override GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IEdge.GetVertex\(Direction\)](#)

See Also

[EventEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdgeIterable Class

A sequence of edges that applies the list of listeners into each edge.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdgeIterable

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventEdgeIterable : ICloseableIterable<IEdge>,
    IDisposable, IEnumerable<IEdge>, IEnumerable
```

VB

```
Public Class EventEdgeIterable
    Implements ICloseableIterable(Of IEdge), IDisposable,
        IEnumerable(Of IEdge), IEnumerable
```

C++


```
public ref class EventEdgeIterable : ICloseableIterable<IEdge^>,
    IDisposable, IEnumerable<IEdge^>, IEnumerable
```

F#



```
type EventEdgeIterable =
    class
        interface ICloseableIterable<IEdge>
        interface IDisposable
        interface IEnumerable<IEdge>
        interface IEnumerable
    end
```

The **EventEdgeIterable** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------|
|  | EventEdgeIterable | Initializes a new instance of the EventEdgeIterable class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|-------------------------------------------------------------|
|  | Dispose | Releases all resources used by the EventEdgeIterable |
|  | GetEnumerator | |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | RelabelEdges | Edges are relabeled by creating new edges with the same properties, but new label. Note that for each edge is deleted and an edge is added. (Defined by EdgeHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdgeIterable Constructor

Initializes a new instance of the [EventEdgeIterable](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventEdgeIterable(  
    IEnumerable<IEdge> iterable,  
    EventGraph eventGraph  
)
```

VB

```
Public Sub New (  
    iterable As IEnumerable(Of IEdge),  
    eventGraph As EventGraph  
)
```

C++

```
public:  
EventEdgeIterable(  
    IEnumerable<IEdge>^ iterable,  
    EventGraph^ eventGraph  
)
```

F#

```
new :  
    iterable : IEnumerable<IEdge> *  
    eventGraph : EventGraph -> EventEdgeIterable
```

Parameters

iterable

Type: [System.Collections.Generic.IEnumerable\(IEdge\)](#)

[Missing <param name="iterable"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdgeIterable.#ctor(System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge},VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

eventGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="eventGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdgeIterable.#ctor(System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IEdge},VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

VelocityDB Class Library

See Also

[EventEdgelterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)


EventEdgeIterable.EventEdgeIterable Methods

The [EventEdgeIterable](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | Dispose | |
|  | GetEnumerator | |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | RelabelEdges | Edges are relabeled by creating new edges with the same properties, but new label. Note that for each edge is deleted and an edge is added. (Defined by EdgeHelpers .) |

See Also

[EventEdgeIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdgeterable.Dispose Method

Releases all resources used by the [EventEdgeterable](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose() sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[EventEdgeterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventEdgelterable.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdgelterable.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^> GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerable<IEdge>  
override GetEnumerator : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdgelterable.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

[EventEdgelterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement Class

An element with a GraphChangedListener attached. Those listeners are notified when changes occur to the properties of the element.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventEdge](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class EventElement : DictionaryElement
```

VB

```
Public MustInherit Class EventElement
    Inherits DictionaryElement
```

C++

```
public ref class EventElement abstract : public DictionaryElement
```

F#



```
[<AbstractClassAttribute>]
type EventElement =
    class
        inherit DictionaryElement
    end
```

The **EventElement** type exposes the following members.








Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object).) |
|  | GetBaseElement | |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  GetHashCode | (Overrides Object.GetHashCode().) |
|  GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  Remove | (Overrides DictionaryElement.Remove().) |
|  RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  ToString | (Overrides Object.ToString().) |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.EventElement Properties

The [EventElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.Id Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also










[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.EventElement Methods

The [EventElement](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object).) |
|  | GetBaseElement | |
|  | GetHashCode | (Overrides Object.GetHashCode().) |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  | Remove | (Overrides DictionaryElement.Remove().) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  | ToString | (Overrides Object.ToString().) |

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.Equals Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.Equals(System.Object)"]

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.GetBaseElement Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetBaseElement"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IElement GetBaseElement ()
```

VB

```
Public Function GetBaseElement As IElement
```

C++

```
public:  
IElement^ GetBaseElement ()
```

F#

```
member GetBaseElement : unit -> IElement
```

Return Value

Type: [IElement](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetBaseElement"]

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.GetHashCode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetHashCode"]

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.GetProperty\(String\)](#)

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.GetPropertyKeys"]

Implements

[IElement.GetPropertyKeys\(\)](#)

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.Remove Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.RemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.RemoveProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.RemoveProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.SetProperty(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.SetProperty(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.SetProperty(System.String, System.Object)"]

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventElement.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement.ToString"]

See Also

[EventElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph Class

An eventInnerTinkerGraph is a wrapper to existing Graph implementations and provides for graph events to be raised to one or more listeners on changes to the Graph. Notifications to the listeners occur for the following events: new vertex/edge, vertex/edge property changed, vertex/edge property removed, vertex/edge removed. The limiting factor to events being raised is related to out-of-process functions changing graph elements. To gather events from eventInnerTinkerGraph, simply provide an implementation of the {@link GraphChangeListener} to the eventInnerTinkerGraph by utilizing the addListener method. eventInnerTinkerGraph allows the addition of multiple GraphChangeListener implementations. Each listener will be notified in the order that it was added.

Inheritance Hierarchy

System.Object

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventGraph : IGraph, IWrapperGraph,
    IDisposable
```

VB

```
Public Class EventGraph
    Implements IGraph, IWrapperGraph, IDisposable
```

C++


```
public ref class EventGraph : IGraph,
    IWrapperGraph, IDisposable
```

F#


```
type EventGraph =
    class
        interface IGraph
        interface IWrapperGraph
        interface IDisposable
    end
```

The **EventGraph** type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------|
|  | EventGraph | Initializes a new instance of the EventGraph class |


Properties













| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|------------------------------------------------------|
|  | AddEdge | |
|  | AddListener | |
|  | AddVertex | |
|  | Dispose | Releases all resources used by the EventGraph |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetListenerIterator | |
|  | GetTrigger | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveAllListeners | |
|  | RemoveEdge | |
|  | RemoveListener | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers.) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  CreateTinkerGraph | (Defined by GraphHelpers.) |
|  GraphString | (Defined by StringFactory.) |
|  LoadGml | (Defined by GraphHelpers.) |
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph Constructor

Initializes a new instance of the [EventGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventGraph(  
    IGraph baseGraph  
)
```

VB

```
Public Sub New (  
    baseGraph As IGraph  
)
```

C++

```
public:  
EventGraph(  
    IGraph^ baseGraph  
)
```

F#

```
new :  
    baseGraph : IGraph -> EventGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="baseGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also


[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.EventGraph Properties

The [EventGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)





EventGraph.EventGraph Methods










The [EventGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddListener | |
|  | AddVertex | |
|  | Dispose | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetListenerIterator | |
|  | GetTrigger | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveAllListeners | |
|  | RemoveEdge | |
|  | RemoveListener | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.AddListener Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddListener(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddListener(  
    IGraphChangedListener listener  
)
```

VB

```
Public Sub AddListener (  
    listener As IGraphChangedListener  
)
```

C++

```
public:  
void AddListener(  
    IGraphChangedListener^ listener  
)
```

F#

```
member AddListener :  
    listener : IGraphChangedListener -> unit
```

Parameters

listener

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener](#)

[Missing <param name="listener"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddListener(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener)"]

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.AddVertex Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.AddVertex(System.Object)"]

Implements

[IGraph.AddVertex\(Object\)](#)

VelocityDB Class Library

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.Dispose Method

Releases all resources used by the [EventGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose() sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdge(System.Object)"]

Implements

[IGraph.GetEdge\(Object\)](#)

VelocityDB Class Library

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetEdges Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable (Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[EventGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[EventGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetListenerIterator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetListenerIterator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IGraphChangedListener> GetListenerIterator()
```

VB

```
Public Function GetListenerIterator As IEnumerable(Of IGraphChangedListener)
```

C++

```
public:  
IEnumerable<IGraphChangedListener^> GetListenerIterator()
```

F#

```
member GetListenerIterator : unit -> IEnumerable<IGraphChangedListener>
```

Return Value

Type: [IEnumerable<IGraphChangedListener>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetListenerIterator"]

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetTrigger Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetTrigger"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventTrigger GetTrigger()
```

VB

```
Public Function GetTrigger As EventTrigger
```

C++

```
public:  
EventTrigger^ GetTrigger()
```

F#

```
member GetTrigger : unit -> EventTrigger
```

Return Value

Type: [EventTrigger](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetTrigger"]

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertex(System.Object)"]

Implements

[IGraph.GetVertex\(Object\)](#)

VelocityDB Class Library

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[EventGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
override GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[EventGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.RemoveAllListeners Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveAllListeners"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveAllListeners ()
```

VB

```
Public Sub RemoveAllListeners
```

C++

```
public:  
void RemoveAllListeners ()
```

F#

```
member RemoveAllListeners : unit -> unit
```

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Public Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void RemoveEdge (  
    IEdge^ edge  
) sealed
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit  
override RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.RemoveListener Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveListener(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveListener (  
    IGraphChangedListener listener  
)
```

VB

```
Public Sub RemoveListener (  
    listener As IGraphChangedListener  
)
```

C++

```
public:  
void RemoveListener (  
    IGraphChangedListener^ listener  
)
```

F#

```
member RemoveListener :  
    listener : IGraphChangedListener -> unit
```

Parameters

listener

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener](#)

[Missing <param name="listener"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveListener(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener)"]

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph.ToString"]

See Also

[EventGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex Class

An index that wraps graph elements in the "evented" way. This class does not directly raise graph events, but passes the GraphChangedListener to the edges and vertices returned from indices so that they may raise graph events.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventIndex : IIndex
```

VB

```
Public Class EventIndex
    Implements IIndex
```

C++


```
public ref class EventIndex : IIndex
```

F#



```
type EventIndex =
    class
        interface IIndex
    end
```

The **EventIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------|-----------------------------------------------------------|
|  | EventIndex | Initializes a new instance of the EventIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex Constructor

Initializes a new instance of the [EventIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventIndex(  
    IIndex rawIndex,  
    EventGraph eventGraph  
)
```

VB

```
Public Sub New (  
    rawIndex As IIndex,  
    eventGraph As EventGraph  
)
```

C++

```
public:  
EventIndex(  
    IIndex^ rawIndex,  
    EventGraph^ eventGraph  
)
```

F#

```
new :  
    rawIndex : IIndex *  
    eventGraph : EventGraph -> EventIndex
```

Parameters

rawIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="rawIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

eventGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="eventGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]



See Also

[EventIndex Class](#)

EventIndex.EventIndex Properties

The [EventIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Name Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Type Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.EventIndex Methods

The [EventIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Count Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Count(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Count(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Count(System.String,System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Count(System.String,System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Query(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Query(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Query(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Query(System.String,System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.Remove Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndex.ToString"]

See Also

[EventIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph Class

EventIndexableGraph is merely a proxy to index methods exposing eventInnerTinkerGraph methods in the "evented" way. Like the eventInnerTinkerGraph it extends from, this graph implementations raise notifications to the listeners for the following events: new vertex/edge, vertex/edge property changed, vertex/edge property removed, vertex/edge removed.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalIndexableGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventIndexableGraph : EventGraph,
    IIndexableGraph, IGraph
```

VB

```
Public Class EventIndexableGraph
    Inherits EventGraph
    Implements IIndexableGraph, IGraph
```

C++


```
public ref class EventIndexableGraph : public EventGraph,
    IIndexableGraph, IGraph
```

F#





```
type EventIndexableGraph =
    class
        inherit EventGraph
        interface IIndexableGraph
        interface IGraph
    end
```

The **EventIndexableGraph** type exposes the following members.















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------|
|  | EventIndexableGraph | Initializes a new instance of the EventIndexableGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph Constructor

Initializes a new instance of the [EventIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventIndexableGraph(  
    IIndexableGraph baseIndexableGraph  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph  
)
```

C++

```
public:  
EventIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph -> EventIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph)"]

See Also





[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)














EventIndexableGraph.EventIndexableGraph Methods

The [EventIndexableGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|
|  | SaveGraphson | (Defined by GraphHelpers.) |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|

See Also

[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph.CreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph.DropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.DropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropIndex(  
    string indexName  
)
```

VB

```
Public Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
) sealed
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.DropIndex(System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
) sealed
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndex(System.String,System.Type)"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndex(System.String,System.Type)"]

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventIndexableGraph.GetIndices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Function GetIndices As IEnumerable(Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndex^>^ GetIndices () sealed
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph.GetIndices"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[EventIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalGraph Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalGraph"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventTransactionalGraph : EventGraph,
    ITransactionalGraph, IGraph
```

VB

```
Public Class EventTransactionalGraph
    Inherits EventGraph
    Implements ITransactionalGraph, IGraph
```

C++


```
public ref class EventTransactionalGraph : public EventGraph,
    ITransactionalGraph, IGraph
```

F#



```
type EventTransactionalGraph =
    class
        inherit EventGraph
        interface ITransactionalGraph
        interface IGraph
    end
```

The **EventTransactionalGraph** type exposes the following members.














Constructors

| Name | Description |
|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  EventTransactionalGraph | Initializes a new instance of the EventTransactionalGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Commit | A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset. |
|  | Rollback | A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalGraph Constructor

Initializes a new instance of the [EventTransactionalGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventTransactionalGraph(  
    ITransactionalGraph transactionalGraph  
)
```

VB

```
Public Sub New (  
    transactionalGraph As ITransactionalGraph  
)
```

C++

```
public:  
EventTransactionalGraph(  
    ITransactionalGraph^ transactionalGraph  
)
```

F#

```
new :  
    transactionalGraph : ITransactionalGraph -> EventTransactionalGraph
```

Parameters

transactionalGraph

Type: [VelocityGraph.Frontenac.Blueprints.ITransactionalGraph](#)

[Missing <param name="transactionalGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalGraph.#ctor(VelocityGraph.Frontenac.Blueprints.ITransactionalGraph)"]

See Also



[EventTransactionalGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)














EventTransactionalGraph.EventTransactionalGraph Methods

The [EventTransactionalGraph](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Commit | A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset. |
|  Rollback | A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[EventTransactionalGraph Class](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalGraph.Commit Method

A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
virtual void Commit () sealed
```

F#

```
abstract Commit : unit -> unit  
override Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[EventTransactionalGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalGraph.Rollback Method

A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
virtual void Rollback() sealed
```

F#

```
abstract Rollback : unit -> unit  
override Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[EventTransactionalGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalIndexableGraph Class

The transactional and indexable implementation of eventInnerTinkerGraph where events are raised in batch in the order they changes occurred to the graph, but only after a successful commit to the underlying graph.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventIndexableGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalIndexableGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventTransactionalIndexableGraph : EventIndexableGraph,
    ITransactionalGraph, IGraph
```

VB

```
Public Class EventTransactionalIndexableGraph
    Inherits EventIndexableGraph
    Implements ITransactionalGraph, IGraph
```

C++


```
public ref class EventTransactionalIndexableGraph : public
EventIndexableGraph,
    ITransactionalGraph, IGraph
```

F#



```
type EventTransactionalIndexableGraph =
    class
        inherit EventIndexableGraph
        interface ITransactionalGraph
        interface IGraph
    end
```

The **EventTransactionalIndexableGraph** type exposes the following members.














Constructors

| Name | Description |
|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
|  EventTransactionalIndexableGraph | Initializes a new instance of the EventTransactionalIndexableGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | Commit | A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset. |
|  | Rollback | A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset. |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalIndexableGraph Constructor

Initializes a new instance of the [EventTransactionalIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventTransactionalIndexableGraph(  
    IIndexableGraph baseIndexableGraph  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph  
)
```

C++

```
public:  
EventTransactionalIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph ->  
EventTransactionalIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTransactionalIndexableGraph.#ctor (VelocityGraph.Frontenac.Blueprints.IIndexableGraph)"]

See Also



[EventTransactionalIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)














EventTransactionalIndexableGraph.EventTransactionalIndexableGraph Methods

The [EventTransactionalIndexableGraph](#) type exposes the following members.

Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  Commit | A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset. |
|  Rollback | A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset. |

Extension Methods

| Name | Description |
|------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutation calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

VelocityDB Class Library

See Also

[EventTransactionalIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalIndexableGraph.Commit Method

A commit only fires the event queue on successful operation. If the commit operation to the underlying graph fails, the event queue will not fire and the queue will not be reset.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
virtual void Commit () sealed
```

F#

```
abstract Commit : unit -> unit  
override Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[EventTransactionalIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTransactionalIndexableGraph.Rollback Method

A rollback only resets the event queue on successful operation. If the rollback operation to the underlying graph fails, the event queue will not be reset.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
virtual void Rollback() sealed
```

F#

```
abstract Rollback : unit -> unit  
override Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[EventTransactionalIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventTrigger : IDisposable
```

VB

```
Public Class EventTrigger
    Implements IDisposable
```

C++


```
public ref class EventTrigger : IDisposable
```

F#





```
type EventTrigger =
    class
        interface IDisposable
    end
```

The **EventTrigger** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | EventTrigger | Initializes a new instance of the EventTrigger class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | AddEvent | Add an event to the event queue. If the enqueueEvents is false, then the queue fires and resets after each event |
|  | Dispose | Releases all resources used by the EventTrigger |
|  | FireEventQueue | |
|  | ResetEventQueue | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger Constructor

Initializes a new instance of the [EventTrigger](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventTrigger(  
    EventGraph graph,  
    bool enqueueEvents  
)
```

VB

```
Public Sub New (  
    graph As EventGraph,  
    enqueueEvents As Boolean  
)
```

C++

```
public:  
EventTrigger(  
    EventGraph^ graph,  
    bool enqueueEvents  
)
```

F#

```
new :  
    graph : EventGraph *  
    enqueueEvents : bool -> EventTrigger
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph,System.Boolean)"]

enqueueEvents

Type: [System.Boolean](#)

[Missing <param name="enqueueEvents"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph,System.Boolean)"]

See Also

[EventTrigger Class](#)





VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger.EventTrigger Methods

The [EventTrigger](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|------------------------------------------------------------------------------------------------------------------|
|  | AddEvent | Add an event to the event queue. If the enqueueEvents is false, then the queue fires and resets after each event |
|  | Dispose | |
|  | FireEventQueue | |
|  | ResetEventQueue | |

See Also

[EventTrigger Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger.AddEvent Method

Add an event to the event queue. If the `enqueueEvents` is false, then the queue fires and resets after each event

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddEvent (
    IEvent evt
)
```

VB

```
Public Sub AddEvent (
    evt As IEvent
)
```

C++

```
public:
void AddEvent (
    IEvent^ evt
)
```

F#

```
member AddEvent :
    evt : IEvent -> unit
```

Parameters

evt

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IEvent](#)

The event to add to the event queue

See Also

[EventTrigger Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger.Dispose Method

Releases all resources used by the [EventTrigger](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose() sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[EventTrigger Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger.FireEventQueue Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger.FireEventQueue"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEventQueue ()
```

VB

```
Public Sub FireEventQueue
```

C++

```
public:  
void FireEventQueue ()
```

F#

```
member FireEventQueue : unit -> unit
```

See Also

[EventTrigger Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventTrigger.ResetEventQueue Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventTrigger.ResetEventQueue"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void ResetEventQueue ()
```

VB

```
Public Sub ResetEventQueue
```

C++

```
public:  
void ResetEventQueue ()
```

F#

```
member ResetEventQueue : unit -> unit
```

See Also

[EventTrigger Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex Class

An vertex with a GraphChangedListener attached. Those listeners are notified when changes occur to the properties of the vertex.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventVertex : EventElement,
    IVertex, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class EventVertex
    Inherits EventElement
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
    KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class EventVertex : public EventElement,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```


F#

```
type EventVertex =
    class
        inherit EventElement
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```
interface ICollection
end
```

The **EventVertex** type exposes the following members.





Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------|
|  | EventVertex | Initializes a new instance of the EventVertex class |







Properties






| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

Methods

| | Name | Description |
|------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex Constructor

Initializes a new instance of the [EventVertex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventVertex(  
    IVertex vertex,  
    EventGraph eventInnerTinkerGraph  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    eventInnerTinkerGraph As EventGraph  
)
```

C++

```
public:  
EventVertex(  
    IVertex^ vertex,  
    EventGraph^ eventInnerTinkerGraph  
)
```

F#

```
new :  
    vertex : IVertex *  
    eventInnerTinkerGraph : EventGraph -> EventVertex
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

eventInnerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="eventInnerTinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

See Also

[EventVertex Class](#)


VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.EventVertex Properties

The [EventVertex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.Vertex Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.Vertex"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex Vertex { get; }
```

VB

```
Public ReadOnly Property Vertex As IVertex  
    Get
```

C++

```
public:  
property IVertex^ Vertex {  
    IVertex^ get ();  
}
```

F#

```
member Vertex : IVertex with get
```

Property Value

Type: [IVertex](#)

See Also





[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)











EventVertex.EventVertex Methods

The [EventVertex](#) type exposes the following members.


Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  | VertexString | (Defined by StringFactory.) |

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    string label,
    IVertex vertex
)
```

VB

```
Public Function AddEdge (
    id As Object,
    label As String,
    vertex As IVertex
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    String^ label,
    IVertex^ vertex
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
override AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Public Function GetVertices (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    Direction direction,  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>  
override GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertex.Query"]

Implements

[IVertex.Query\(\)](#)

See Also

[EventVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertexIterable Class

A sequence of vertices that applies the list of listeners into each vertex.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertexIterable

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EventVertexIterable : ICloseableIterable<IVertex>,
    IDisposable, IEnumerable<IVertex>, IEnumerable
```

VB

```
Public Class EventVertexIterable
    Implements ICloseableIterable(Of IVertex), IDisposable,
        IEnumerable(Of IVertex), IEnumerable
```

C++


```
public ref class EventVertexIterable : ICloseableIterable<IVertex^>,
    IDisposable, IEnumerable<IVertex^>, IEnumerable
```

F#



```
type EventVertexIterable =
    class
        interface ICloseableIterable<IVertex>
        interface IDisposable
        interface IEnumerable<IVertex>
        interface IEnumerable
    end
```

The **EventVertexIterable** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------|
|  | EventVertexIterable | Initializes a new instance of the EventVertexIterable class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------|
|  | Dispose | Releases all resources used by the EventVertexIterable |
|  | GetEnumerator | |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertexIterable Constructor

Initializes a new instance of the [EventVertexIterable](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EventVertexIterable(  
    IEnumerable<IVertex> iterable,  
    EventGraph eventGraph  
)
```

VB

```
Public Sub New (  
    iterable As IEnumerable(Of IVertex),  
    eventGraph As EventGraph  
)
```

C++

```
public:  
EventVertexIterable(  
    IEnumerable<IVertex^>^ iterable,  
    EventGraph^ eventGraph  
)
```

F#

```
new :  
    iterable : IEnumerable<IVertex> *  
    eventGraph : EventGraph -> EventVertexIterable
```

Parameters

iterable

Type: [System.Collections.Generic.IEnumerable<IVertex>](#)

[Missing <param name="iterable"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertexIterable.#ctor(System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex},VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

eventGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph](#)

[Missing <param name="eventGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertexIterable.#ctor(System.Collections.Generic.IEnumerable{VelocityGraph.Frontenac.Blueprints.IVertex},VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventGraph)"]

VelocityDB Class Library

See Also

[EventVertexIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertexIterable.EventVertexIterable Methods

The [EventVertexIterable](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|-------------|
|  | Dispose | |
|  | GetEnumerator | |

See Also

[EventVertexIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertexIterable.Dispose Method

Releases all resources used by the [EventVertexIterable](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Dispose ()
```

VB

```
Public Sub Dispose
```

C++

```
public:  
virtual void Dispose () sealed
```

F#

```
abstract Dispose : unit -> unit  
override Dispose : unit -> unit
```

Implements

[IDisposable.Dispose\(\)](#)

See Also

[EventVertexIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)

EventVertexIterable.GetEnumerator Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertexIterable.GetEnumerator"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerator<IVertex> GetEnumerator()
```

VB

```
Public Function GetEnumerator As IEnumerator(Of IVertex)
```

C++

```
public:  
virtual IEnumerator<IVertex^> GetEnumerator() sealed
```

F#

```
abstract GetEnumerator : unit -> IEnumerator<IVertex>  
override GetEnumerator : unit -> IEnumerator<IVertex>
```

Return Value

Type: [IEnumerator<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.EventVertexIterable.GetEnumerator"]

Implements

[IEnumerable\(T\).GetEnumerator\(\)](#)

See Also

















[EventVertexIterable Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event Namespace](#)



VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener"]

Classes

| Class | Description |
|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
|  ConsoleGraphChangedListener | An example listener that writes a message to the console for each event that fires from the graph. |
|  EdgeAddedEvent | |
|  EdgePropertyChangedEvent | |
|  EdgePropertyEvent | |
|  EdgePropertyEventContract | |
|  EdgePropertyRemovedEvent | Event fired when an edge property is removed. |
|  EdgeRemovedEvent | Event fired when an edge is removed. |
|  EventContract | |
|  GraphChangedListenerContract | |
|  StubGraphChangedListener | An event listener that acts as a counter for changes to the graph. |
|  VertexAddedEvent | Event that fires when a vertex is added to a graph. |
|  VertexPropertyChangedEvent | Event that fires when a property changes on a vertex. |
|  VertexPropertyEvent | Base class for property changed events. |
|  VertexPropertyEventContract | |
|  VertexPropertyRemovedEvent | Event fired when a vertex property is removed. |
|  VertexRemovedEvent | Event fired when a vertex is removed. |

Interfaces

| Interface | Description |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IEvent | An event generated by changes to the graph. |
|  IGraphChangedListener | Interface for a listener to eventInnerTinkerGraph change events. Implementations of this interface should be added to the list of listeners on the addListener method on the eventInnerTinkerGraph. |

ConsoleGraphChangedListener Class

An example listener that writes a message to the console for each event that fires from the graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ConsoleGraphChangedListener : IGraphChangedListener
```

VB

```
Public Class ConsoleGraphChangedListener
    Implements IGraphChangedListener
```

C++


```
public ref class ConsoleGraphChangedListener : IGraphChangedListener
```

F#






```
type ConsoleGraphChangedListener =
    class
        interface IGraphChangedListener
    end
```

The **ConsoleGraphChangedListener** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|----------------------------------------------------------------------------|
|  | ConsoleGraphChangedListener | Initializes a new instance of the ConsoleGraphChangedListener class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|-------------|
|  | EdgeAdded | |
|  | EdgePropertyChanged | |
|  | EdgePropertyRemoved | |
|  | EdgeRemoved | |
|  | VertexAdded | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------------|--|
|  | VertexPropertyChanged | |
|  | VertexPropertyRemoved | |
|  | VertexRemoved | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener Constructor

Initializes a new instance of the [ConsoleGraphChangedListener](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ConsoleGraphChangedListener(  
    IGraph graph  
)
```

VB

```
Public Sub New (  
    graph As IGraph  
)
```

C++

```
public:  
ConsoleGraphChangedListener(  
    IGraph^ graph  
)
```

F#

```
new :  
    graph : IGraph -> ConsoleGraphChangedListener
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="graph"/> documentation for

**"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.
#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]**

See Also









[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.ConsoleGraphChangedListener Methods

The [ConsoleGraphChangedListener](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|-------------|
|  | EdgeAdded | |
|  | EdgePropertyChanged | |
|  | EdgePropertyRemoved | |
|  | EdgeRemoved | |
|  | VertexAdded | |
|  | VertexPropertyChanged | |
|  | VertexPropertyRemoved | |
|  | VertexRemoved | |

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.EdgeAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgeAdded(  
    IEdge edge  
)
```

VB

```
Public Sub EdgeAdded (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void EdgeAdded(  
    IEdge^ edge  
) sealed
```

F#

```
abstract EdgeAdded :  
    edge : IEdge -> unit  
override EdgeAdded :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraphChangedListener.EdgeAdded\(IEdge\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.EdgePropertyChanged Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgePropertyChanged(  
    IEdge edge,  
    string key,  
    Object oldValue,  
    Object newValue  
)
```

VB

```
Public Sub EdgePropertyChanged (  
    edge As IEdge,  
    key As String,  
    oldValue As Object,  
    newValue As Object  
)
```

C++

```
public:  
virtual void EdgePropertyChanged(  
    IEdge^ edge,  
    String^ key,  
    Object^ oldValue,  
    Object^ newValue  
) sealed
```

F#

```
abstract EdgePropertyChanged :  
    edge : IEdge *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit  
override EdgePropertyChanged :  
    edge : IEdge *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

newValue

Type: [System.Object](#)

[Missing <param name="newValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Implements

[IGraphChangeListener.EdgePropertyChanged\(IEdge, String, Object, Object\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.EdgePropertyRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgePropertyRemoved(  
    IEdge edge,  
    string key,  
    Object removedValue  
)
```

VB

```
Public Sub EdgePropertyRemoved (  
    edge As IEdge,  
    key As String,  
    removedValue As Object  
)
```

C++

```
public:  
virtual void EdgePropertyRemoved(  
    IEdge^ edge,  
    String^ key,  
    Object^ removedValue  
) sealed
```

F#

```
abstract EdgePropertyRemoved :  
    edge : IEdge *  
    key : string *  
    removedValue : Object -> unit  
override EdgePropertyRemoved :  
    edge : IEdge *  
    key : string *  
    removedValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

removedValue

Type: [System.Object](#)

[Missing <param name="removedValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

Implements

[IGraphChangedListener.EdgePropertyRemoved\(IEdge, String, Object\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.EdgeRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgeRemoved(  
    IEdge edge,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub EdgeRemoved (  
    edge As IEdge,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
virtual void EdgeRemoved(  
    IEdge^ edge,  
    IDictionary<String^, Object^>^ props  
) sealed
```

F#

```
abstract EdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit  
override EdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

**[Missing <param name="props" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.
EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.
String,System.Object})"]**

Implements

[IGraphChangedListener.EdgeRemoved\(IEdge, IDictionary\(String, Object\)\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.VertexAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexAdded(  
    IVertex vertex  
)
```

VB

```
Public Sub VertexAdded (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void VertexAdded(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract VertexAdded :  
    vertex : IVertex -> unit  
override VertexAdded :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraphChangedListener.VertexAdded\(IVertex\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.VertexPropertyChanged Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexPropertyChanged(  
    IVertex vertex,  
    string key,  
    Object oldValue,  
    Object newValue  
)
```

VB

```
Public Sub VertexPropertyChanged (  
    vertex As IVertex,  
    key As String,  
    oldValue As Object,  
    newValue As Object  
)
```

C++

```
public:  
virtual void VertexPropertyChanged(  
    IVertex^ vertex,  
    String^ key,  
    Object^ oldValue,  
    Object^ newValue  
) sealed
```

F#

```
abstract VertexPropertyChanged :  
    vertex : IVertex *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit  
override VertexPropertyChanged :  
    vertex : IVertex *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

newValue

Type: [System.Object](#)

[Missing <param name="newValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Implements

[IGraphChangedListener.VertexPropertyChanged\(IVertex, String, Object, Object\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.VertexPropertyRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexPropertyRemoved(  
    IVertex vertex,  
    string key,  
    Object removedValue  
)
```

VB

```
Public Sub VertexPropertyRemoved (  
    vertex As IVertex,  
    key As String,  
    removedValue As Object  
)
```

C++

```
public:  
virtual void VertexPropertyRemoved(  
    IVertex^ vertex,  
    String^ key,  
    Object^ removedValue  
) sealed
```

F#

```
abstract VertexPropertyRemoved :  
    vertex : IVertex *  
    key : string *  
    removedValue : Object -> unit  
override VertexPropertyRemoved :  
    vertex : IVertex *  
    key : string *  
    removedValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

removedValue

Type: [System.Object](#)

[Missing <param name="removedValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

Implements

[IGraphChangedListener.VertexPropertyRemoved\(IVertex, String, Object\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

ConsoleGraphChangedListener.VertexRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexRemoved(  
    IVertex vertex,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub VertexRemoved (  
    vertex As IVertex,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
virtual void VertexRemoved(  
    IVertex^ vertex,  
    IDictionary<String^, Object^>^ props  
) sealed
```

F#

```
abstract VertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit  
override VertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

**[Missing <param name="props" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.ConsoleGraphChangedListener.
VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{S
ystem.String,System.Object})"]**

Implements

[IGraphChangedListener.VertexRemoved\(IVertex, IDictionary\(String, Object\)\)](#)

See Also

[ConsoleGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeAddedEvent Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeAddedEvent](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeAddedEvent

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EdgeAddedEvent : IEvent
```

VB

```
Public Class EdgeAddedEvent
    Implements IEvent
```

C++


```
public ref class EdgeAddedEvent : IEvent
```

F#


```
type EdgeAddedEvent =
    class
        interface IEvent
    end
```

The **EdgeAddedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | EdgeAddedEvent | Initializes a new instance of the EdgeAddedEvent class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeAddedEvent Constructor

Initializes a new instance of the [EdgeAddedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeAddedEvent (  
    IEdge edge  
)
```

VB

```
Public Sub New (  
    edge As IEdge  
)
```

C++

```
public:  
EdgeAddedEvent (  
    IEdge^ edge  
)
```

F#

```
new :  
    edge : IEdge -> EdgeAddedEvent
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeAddedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge)"]

See Also


[EdgeAddedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeAddedEvent.EdgeAddedEvent Methods

The [EdgeAddedEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[EdgeAddedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeAddedEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeAddedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
virtual void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
) sealed
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
override FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeAddedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerator\(IGraphChangedListener\)\)](#)

See Also

[EdgeAddedEvent Class](#)

EdgePropertyChangedEvent Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EdgePropertyChangedEvent : EdgePropertyEvent
```

VB

```
Public Class EdgePropertyChangedEvent  
    Inherits EdgePropertyEvent
```

C++


```
public ref class EdgePropertyChangedEvent : public EdgePropertyEvent
```

F#

```
type EdgePropertyChangedEvent =  
    class  
        inherit EdgePropertyEvent  
    end
```

The **EdgePropertyChangedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|-------------------------------------------------------------------------|
|  | EdgePropertyChangedEvent | Initializes a new instance of the EdgePropertyChangedEvent class |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyChangedEvent Constructor

Initializes a new instance of the [EdgePropertyChangedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgePropertyChangedEvent (
    IEdge edge,
    string key,
    Object oldValue,
    Object newValue
)
```

VB

```
Public Sub New (
    edge As IEdge,
    key As String,
    oldValue As Object,
    newValue As Object
)
```

C++

```
public:
EdgePropertyChangedEvent (
    IEdge^ edge,
    String^ key,
    Object^ oldValue,
    Object^ newValue
)
```

F#

```
new :
    edge : IEdge *
    key : string *
    oldValue : Object *
    newValue : Object -> EdgePropertyChangedEvent
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

newValue

Type: [System.Object](#)

[Missing <param name="newValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

See Also

[EdgePropertyChangedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyEvent Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyChangedEvent](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyRemovedEvent](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class EdgePropertyEvent : IEvent
```

VB

```
Public MustInherit Class EdgePropertyEvent  
    Implements IEvent
```

C++


```
public ref class EdgePropertyEvent abstract : IEvent
```

F#

```
[<AbstractClassAttribute>]  
type EdgePropertyEvent =  
    class  
        interface IEvent  
    end
```

The **EdgePropertyEvent** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyEvent.EdgePropertyEvent Methods

The [EdgePropertyEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[EdgePropertyEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
virtual void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
) sealed
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
override FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerable\(IGraphChangedListener\)\)](#)

See Also

[EdgePropertyEvent Class](#)

EdgePropertyEventContract Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class EdgePropertyEventContract
```

VB

```
Public NotInheritable Class EdgePropertyEventContract
```

C++


```
public ref class EdgePropertyEventContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type EdgePropertyEventContract = class end
```

The **EdgePropertyEventContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | ValidateFire | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyEventContract.EdgePropertyEventContract Methods

The [EdgePropertyEventContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | ValidateFire | |

See Also

[EdgePropertyEventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyEventContract.ValidateFire Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateFire(  
    IGraphChangedListener listener,  
    IEdge edge,  
    string key,  
    Object oldValue,  
    Object newValue  
)
```

VB

```
Public Shared Sub ValidateFire (  
    listener As IGraphChangedListener,  
    edge As IEdge,  
    key As String,  
    oldValue As Object,  
    newValue As Object  
)
```

C++

```
public:  
static void ValidateFire(  
    IGraphChangedListener^ listener,  
    IEdge^ edge,  
    String^ key,  
    Object^ oldValue,  
    Object^ newValue  
)
```

F#

```
static member ValidateFire :  
    listener : IGraphChangedListener *  
    edge : IEdge *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit
```

Parameters

listener

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener](#)

[Missing <param name="listener"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

newValue

Type: [System.Object](#)

[Missing <param name="newValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener,VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

See Also

[EdgePropertyEventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyRemovedEvent Class

Event fired when an edge property is removed.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyEvent](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyRemovedEvent](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EdgePropertyRemovedEvent : EdgePropertyEvent
```

VB

```
Public Class EdgePropertyRemovedEvent  
    Inherits EdgePropertyEvent
```

C++


```
public ref class EdgePropertyRemovedEvent : public EdgePropertyEvent
```

F#

```
type EdgePropertyRemovedEvent =  
    class  
        inherit EdgePropertyEvent  
    end
```

The **EdgePropertyRemovedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|-------------------------------------------------------------------------|
|  | EdgePropertyRemovedEvent | Initializes a new instance of the EdgePropertyRemovedEvent class |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgePropertyRemovedEvent Constructor

Initializes a new instance of the [EdgePropertyRemovedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgePropertyRemovedEvent (  
    IEdge vertex,  
    string key,  
    Object oldValue  
)
```

VB

```
Public Sub New (  
    vertex As IEdge,  
    key As String,  
    oldValue As Object  
)
```

C++

```
public:  
EdgePropertyRemovedEvent (  
    IEdge^ vertex,  
    String^ key,  
    Object^ oldValue  
)
```

F#

```
new :  
    vertex : IEdge *  
    key : string *  
    oldValue : Object -> EdgePropertyRemovedEvent
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgePropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

See Also

[EdgePropertyRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeRemovedEvent Class

Event fired when an edge is removed.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeRemovedEvent

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class EdgeRemovedEvent : IEvent
```

VB

```
Public Class EdgeRemovedEvent  
    Implements IEvent
```

C++


```
public ref class EdgeRemovedEvent : IEvent
```

F#


```
type EdgeRemovedEvent =  
    class  
        interface IEvent  
    end
```

The **EdgeRemovedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
|  | EdgeRemovedEvent | Initializes a new instance of the EdgeRemovedEvent class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeRemovedEvent Constructor

Initializes a new instance of the [EdgeRemovedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public EdgeRemovedEvent (  
    IEdge edge,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub New (  
    edge As IEdge,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
EdgeRemovedEvent (  
    IEdge^ edge,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
new :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> EdgeRemovedEvent
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

[Missing <param name="props"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

VelocityDB Class Library

See Also


[EdgeRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeRemovedEvent.EdgeRemovedEvent Methods

The [EdgeRemovedEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[EdgeRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EdgeRemovedEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeRemovedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
virtual void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
) sealed
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
override FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EdgeRemovedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerator\(IGraphChangedListener\)\)](#)

See Also

[EdgeRemovedEvent Class](#)

EventContract Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EventContract](#)"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EventContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class EventContract
```

VB

```
Public NotInheritable Class EventContract
```

C++

```
public ref class EventContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type EventContract = class end
```

The **EventContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | ValidateFireEvent | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EventContract.EventContract Methods

The [EventContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------|-------------|
|  | ValidateFireEvent | |

See Also

[EventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

EventContract.ValidateFireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EventContract.ValidateFireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateFireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Shared Sub ValidateFireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
static void ValidateFireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
)
```

F#

```
static member ValidateFireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.EventContract.ValidateFireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

See Also

[EventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class GraphChangedListenerContract
```

VB

```
Public NotInheritable Class GraphChangedListenerContract
```

C++









```
public ref class GraphChangedListenerContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type GraphChangedListenerContract = class end
```

The **GraphChangedListenerContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------|-------------|
|  | ValidateEdgeAdded | |
|  | ValidateEdgePropertyChanged | |
|  | ValidateEdgePropertyRemoved | |
|  | ValidateEdgeRemoved | |
|  | ValidateVertexAdded | |
|  | ValidateVertexPropertyChanged | |
|  | ValidateVertexPropertyRemoved | |
|  | ValidateVertexRemoved | |

VelocityDB Class Library









See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.GraphChangedListenerContract Methods

The [GraphChangedListenerContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------|-------------|
|  | ValidateEdgeAdded | |
|  | ValidateEdgePropertyChanged | |
|  | ValidateEdgePropertyRemoved | |
|  | ValidateEdgeRemoved | |
|  | ValidateVertexAdded | |
|  | ValidateVertexPropertyChanged | |
|  | ValidateVertexPropertyRemoved | |
|  | ValidateVertexRemoved | |

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateEdgeAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateEdgeAdded(  
    IEdge edge  
)
```

VB

```
Public Shared Sub ValidateEdgeAdded (  
    edge As IEdge  
)
```

C++

```
public:  
static void ValidateEdgeAdded(  
    IEdge^ edge  
)
```

F#

```
static member ValidateEdgeAdded :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateEdgePropertyChanged Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateEdgePropertyChanged (
    IEdge edge,
    string key,
    Object oldValue,
    Object setValue
)
```

VB

```
Public Shared Sub ValidateEdgePropertyChanged (
    edge As IEdge,
    key As String,
    oldValue As Object,
    setValue As Object
)
```

C++

```
public:
static void ValidateEdgePropertyChanged (
    IEdge^ edge,
    String^ key,
    Object^ oldValue,
    Object^ setValue
)
```

F#

```
static member ValidateEdgePropertyChanged :
    edge : IEdge *
    key : string *
    oldValue : Object *
    setValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.

ValidateEdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

setValue

Type: [System.Object](#)

[Missing <param name="setValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateEdgePropertyRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateEdgePropertyRemoved(  
    IEdge edge,  
    string key,  
    Object removedValue  
)
```

VB

```
Public Shared Sub ValidateEdgePropertyRemoved (  
    edge As IEdge,  
    key As String,  
    removedValue As Object  
)
```

C++

```
public:  
static void ValidateEdgePropertyRemoved(  
    IEdge^ edge,  
    String^ key,  
    Object^ removedValue  
)
```

F#

```
static member ValidateEdgePropertyRemoved :  
    edge : IEdge *  
    key : string *  
    removedValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

removedValue

Type: [System.Object](#)

[Missing <param name="removedValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateEdgeRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateEdgeRemoved(  
    IEdge edge,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Shared Sub ValidateEdgeRemoved (  
    edge As IEdge,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
static void ValidateEdgeRemoved(  
    IEdge^ edge,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
static member ValidateEdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateEdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary](#)([String](#), [Object](#))

**[Missing <param name="props"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.
ValidateEdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDiction
ary{System.String,System.Object})"]**

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateVertexAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateVertexAdded(  
    IVertex vertex  
)
```

VB

```
Public Shared Sub ValidateVertexAdded (  
    vertex As IVertex  
)
```

C++

```
public:  
static void ValidateVertexAdded(  
    IVertex^ vertex  
)
```

F#

```
static member ValidateVertexAdded :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateVertexPropertyChanged Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateVertexPropertyChanged (
    IVertex vertex,
    string key,
    Object oldValue,
    Object setValue
)
```

VB

```
Public Shared Sub ValidateVertexPropertyChanged (
    vertex As IVertex,
    key As String,
    oldValue As Object,
    setValue As Object
)
```

C++

```
public:
static void ValidateVertexPropertyChanged (
    IVertex^ vertex,
    String^ key,
    Object^ oldValue,
    Object^ setValue
)
```

F#

```
static member ValidateVertexPropertyChanged :
    vertex : IVertex *
    key : string *
    oldValue : Object *
    setValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

setValue

Type: [System.Object](#)

[Missing <param name="setValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateVertexPropertyRemoved Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateVertexPropertyRemoved (
    IVertex vertex,
    string key,
    Object removedValue
)
```

VB

```
Public Shared Sub ValidateVertexPropertyRemoved (
    vertex As IVertex,
    key As String,
    removedValue As Object
)
```

C++

```
public:
static void ValidateVertexPropertyRemoved (
    IVertex^ vertex,
    String^ key,
    Object^ removedValue
)
```

F#

```
static member ValidateVertexPropertyRemoved :
    vertex : IVertex *
    key : string *
    removedValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.

ValidateVertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

removedValue

Type: [System.Object](#)

[Missing <param name="removedValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

GraphChangedListenerContract.ValidateVertexRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateVertexRemoved(  
    IVertex vertex,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Shared Sub ValidateVertexRemoved (  
    vertex As IVertex,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
static void ValidateVertexRemoved(  
    IVertex^ vertex,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
static member ValidateVertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.ValidateVertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary](#)([String](#), [Object](#))

**[Missing <param name="props"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.GraphChangedListenerContract.
ValidateVertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDict
ionary{System.String,System.Object})"]**

See Also

[GraphChangedListenerContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IEvent Interface

An event generated by changes to the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IEvent
```

VB

```
Public Interface IEvent
```

C++


```
public interface class IEvent
```

F#

```
type IEvent = interface end
```

The **IEvent** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IEvent.IEvent Methods

The [IEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[IEvent Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
)
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

See Also

[IEvent Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener Interface

Interface for a listener to eventInnerTinkerGraph change events. Implementations of this interface should be added to the list of listeners on the addListener method on the eventInnerTinkerGraph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IGraphChangedListener
```

VB

```
Public Interface IGraphChangedListener
```

C++









```
public interface class IGraphChangedListener
```

F#

```
type IGraphChangedListener = interface end
```

The **IGraphChangedListener** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------------|
|  | EdgeAdded | Raised after a new edge is added. |
|  | EdgePropertyChanged | Raised after the property of a edge changed. |
|  | EdgePropertyRemoved | Raised after an edge property was removed. |
|  | EdgeRemoved | Raised after an edge was removed from the graph. |
|  | VertexAdded | Raised when a new Vertex is added. |
|  | VertexPropertyChanged | Raised after the property of a vertex changed. |
|  | VertexPropertyRemoved | Raised after a vertex property was removed. |
|  | VertexRemoved | Raised after a vertex was removed from the graph. |









See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.IGraphChangedListener Methods

The [IGraphChangedListener](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------------|
|  | EdgeAdded | Raised after a new edge is added. |
|  | EdgePropertyChanged | Raised after the property of a edge changed. |
|  | EdgePropertyRemoved | Raised after an edge property was removed. |
|  | EdgeRemoved | Raised after an edge was removed from the graph. |
|  | VertexAdded | Raised when a new Vertex is added. |
|  | VertexPropertyChanged | Raised after the property of a vertex changed. |
|  | VertexPropertyRemoved | Raised after a vertex property was removed. |
|  | VertexRemoved | Raised after a vertex was removed from the graph. |

See Also

[IGraphChangedListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.EdgeAdded Method

Raised after a new edge is added.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void EdgeAdded (
    IEdge edge
)
```

VB

```
Sub EdgeAdded (
    edge As IEdge
)
```

C++

```
void EdgeAdded (
    IEdge^ edge
)
```

F#

```
abstract EdgeAdded :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener.EdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

See Also

[IGraphChangedListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangeListener.EdgePropertyChanged Method

Raised after the property of a edge changed.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void EdgePropertyChanged(  
    IEdge edge,  
    string key,  
    Object oldValue,  
    Object setValue  
)
```

VB

```
Sub EdgePropertyChanged (  
    edge As IEdge,  
    key As String,  
    oldValue As Object,  
    setValue As Object  
)
```

C++

```
void EdgePropertyChanged(  
    IEdge^ edge,  
    String^ key,  
    Object^ oldValue,  
    Object^ setValue  
)
```

F#

```
abstract EdgePropertyChanged :  
    edge : IEdge *  
    key : string *  
    oldValue : Object *  
    setValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the edge that changed

key

Type: [System.String](#)

the key of the property that changed

VelocityDB Class Library

oldValue

Type: [System.Object](#)

the old value of the property

setValue

Type: [System.Object](#)

the new value of the property

See Also

[IGraphChangeListener Interface](#)

[VelocityGraph.Frontend.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.EdgePropertyRemoved Method

Raised after an edge property was removed.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void EdgePropertyRemoved(  
    IEdge edge,  
    string key,  
    Object removedValue  
)
```

VB

```
Sub EdgePropertyRemoved (  
    edge As IEdge,  
    key As String,  
    removedValue As Object  
)
```

C++

```
void EdgePropertyRemoved(  
    IEdge^ edge,  
    String^ key,  
    Object^ removedValue  
)
```

F#

```
abstract EdgePropertyRemoved :  
    edge : IEdge *  
    key : string *  
    removedValue : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the edge that changed

key

Type: [System.String](#)

the key that was removed

removedValue

Type: [System.Object](#)

the value of the property that was removed

VelocityDB Class Library

See Also

[IGraphChangeListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.EdgeRemoved Method

Raised after an edge was removed from the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void EdgeRemoved(  
    IEdge edge,  
    IDictionary<string, Object> props  
)
```

VB

```
Sub EdgeRemoved (  
    edge As IEdge,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
void EdgeRemoved(  
    IEdge^ edge,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
abstract EdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

the edge that was removed

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

[Missing <param name="props"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

See Also

[IGraphChangedListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.VertexAdded Method

Raised when a new Vertex is added.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void VertexAdded(  
    IVertex vertex  
)
```

VB

```
Sub VertexAdded (  
    vertex As IVertex  
)
```

C++

```
void VertexAdded(  
    IVertex^ vertex  
)
```

F#

```
abstract VertexAdded :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex that was added

See Also

[IGraphChangedListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangeListener.VertexPropertyChanged Method

Raised after the property of a vertex changed.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void VertexPropertyChanged(  
    IVertex vertex,  
    string key,  
    Object oldValue,  
    Object setValue  
)
```

VB

```
Sub VertexPropertyChanged (  
    vertex As IVertex,  
    key As String,  
    oldValue As Object,  
    setValue As Object  
)
```

C++

```
void VertexPropertyChanged(  
    IVertex^ vertex,  
    String^ key,  
    Object^ oldValue,  
    Object^ setValue  
)
```

F#

```
abstract VertexPropertyChanged :  
    vertex : IVertex *  
    key : string *  
    oldValue : Object *  
    setValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex that changed

key

Type: [System.String](#)

the key of the property that changed

VelocityDB Class Library

oldValue

Type: [System.Object](#)

the old value of the property

setValue

Type: [System.Object](#)

the new value of the property

See Also

[IGraphChangeListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.VertexPropertyRemoved Method

Raised after a vertex property was removed.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void VertexPropertyRemoved(  
    IVertex vertex,  
    string key,  
    Object removedValue  
)
```

VB

```
Sub VertexPropertyRemoved (  
    vertex As IVertex,  
    key As String,  
    removedValue As Object  
)
```

C++

```
void VertexPropertyRemoved(  
    IVertex^ vertex,  
    String^ key,  
    Object^ removedValue  
)
```

F#

```
abstract VertexPropertyRemoved :  
    vertex : IVertex *  
    key : string *  
    removedValue : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex that changed

key

Type: [System.String](#)

the key that was removed

removedValue

Type: [System.Object](#)

the value of the property that was removed

VelocityDB Class Library

See Also

[IGraphChangeListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

IGraphChangedListener.VertexRemoved Method

Raised after a vertex was removed from the graph.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
void VertexRemoved(  
    IVertex vertex,  
    IDictionary<string, Object> props  
)
```

VB

```
Sub VertexRemoved (  
    vertex As IVertex,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
void VertexRemoved(  
    IVertex^ vertex,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
abstract VertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

the vertex that was removed

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

[Missing <param name="props"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

See Also

[IGraphChangedListener Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener Class

An event listener that acts as a counter for changes to the graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class StubGraphChangedListener : IGraphChangedListener
```

VB

```
Public Class StubGraphChangedListener
    Implements IGraphChangedListener
```

C++


```
public ref class StubGraphChangedListener : IGraphChangedListener
```

F#







```
type StubGraphChangedListener =
    class
        interface IGraphChangedListener
    end
```

The **StubGraphChangedListener** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|-------------------------------------------------------------------------|
|  | StubGraphChangedListener | Initializes a new instance of the StubGraphChangedListener class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------------|-------------|
|  | AddEdgeEventRecorded | |
|  | AddVertexEventRecorded | |
|  | EdgeAdded | |
|  | EdgePropertyChanged | |
|  | EdgePropertyChangedEventRecorded | |
|  | EdgePropertyRemoved | |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|----------------------------------------------------|--|
|  | EdgePropertyRemovedEventRecorded | |
|  | EdgeRemoved | |
|  | EdgeRemovedEventRecorded | |
|  | GetOrder | |
|  | Reset | |
|  | VertexAdded | |
|  | VertexPropertyChanged | |
|  | VertexPropertyChangedEventRecorded | |
|  | VertexPropertyRemoved | |
|  | VertexPropertyRemovedEventRecorded | |
|  | VertexRemoved | |
|  | VertexRemovedEventRecorded | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener Constructor

Initializes a new instance of the [StubGraphChangeListener](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public StubGraphChangeListener ()
```

VB

```
Public Sub New
```

C++

```
public:  
StubGraphChangeListener ()
```

F#

```
new : unit -> StubGraphChangeListener
```

See Also



















[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.StubGraphChangeListener Methods

The [StubGraphChangeListener](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------------------------|-------------|
|  | AddEdgeEventRecorded | |
|  | AddVertexEventRecorded | |
|  | EdgeAdded | |
|  | EdgePropertyChanged | |
|  | EdgePropertyChangedEventRecorded | |
|  | EdgePropertyRemoved | |
|  | EdgePropertyRemovedEventRecorded | |
|  | EdgeRemoved | |
|  | EdgeRemovedEventRecorded | |
|  | GetOrder | |
|  | Reset | |
|  | VertexAdded | |
|  | VertexPropertyChanged | |
|  | VertexPropertyChangedEventRecorded | |
|  | VertexPropertyRemoved | |
|  | VertexPropertyRemovedEventRecorded | |
|  | VertexRemoved | |
|  | VertexRemovedEventRecorded | |

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.AddEdgeEventRecorded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.AddEdgeEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long AddEdgeEventRecorded()
```

VB

```
Public Function AddEdgeEventRecorded As Long
```

C++

```
public:  
long long AddEdgeEventRecorded()
```

F#

```
member AddEdgeEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.AddEdgeEventRecorded"]

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.AddVertexEventRecorded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.AddVertexEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long AddVertexEventRecorded()
```

VB

```
Public Function AddVertexEventRecorded As Long
```

C++

```
public:  
long long AddVertexEventRecorded()
```

F#

```
member AddVertexEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.AddVertexEventRecorded"]

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.EdgeAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgeAdded (  
    IEdge edge  
)
```

VB

```
Public Sub EdgeAdded (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void EdgeAdded (  
    IEdge^ edge  
) sealed
```

F#

```
abstract EdgeAdded :  
    edge : IEdge -> unit  
override EdgeAdded :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgeAdded(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraphChangedListener.EdgeAdded\(IEdge\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.EdgePropertyChanged Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgePropertyChanged(  
    IEdge edge,  
    string s,  
    Object o,  
    Object n  
)
```

VB

```
Public Sub EdgePropertyChanged (  
    edge As IEdge,  
    s As String,  
    o As Object,  
    n As Object  
)
```

C++

```
public:  
virtual void EdgePropertyChanged(  
    IEdge^ edge,  
    String^ s,  
    Object^ o,  
    Object^ n  
) sealed
```

F#

```
abstract EdgePropertyChanged :  
    edge : IEdge *  
    s : string *  
    o : Object *  
    n : Object -> unit  
override EdgePropertyChanged :  
    edge : IEdge *  
    s : string *  
    o : Object *  
    n : Object -> unit
```


Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

s

Type: [System.String](#)

[Missing <param name="s"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

o

Type: [System.Object](#)

[Missing <param name="o"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

n

Type: [System.Object](#)

[Missing <param name="n"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChanged(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object,System.Object)"]

Implements

[IGraphChangeListener.EdgePropertyChanged\(IEdge, String, Object, Object\)](#)

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.EdgePropertyChangedEventRecorded Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChangedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long EdgePropertyChangedEventRecorded()
```

VB

```
Public Function EdgePropertyChangedEventRecorded As Long
```

C++

```
public:  
long long EdgePropertyChangedEventRecorded()
```

F#

```
member EdgePropertyChangedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyChangedEventRecorded"]

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.EdgePropertyRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgePropertyRemoved(  
    IEdge edge,  
    string s,  
    Object o  
)
```

VB

```
Public Sub EdgePropertyRemoved (  
    edge As IEdge,  
    s As String,  
    o As Object  
)
```

C++

```
public:  
virtual void EdgePropertyRemoved(  
    IEdge^ edge,  
    String^ s,  
    Object^ o  
) sealed
```

F#

```
abstract EdgePropertyRemoved :  
    edge : IEdge *  
    s : string *  
    o : Object -> unit  
override EdgePropertyRemoved :  
    edge : IEdge *  
    s : string *  
    o : Object -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

s

Type: [System.String](#)

[Missing <param name="s"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

o

Type: [System.Object](#)

[Missing <param name="o"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgePropertyRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.String,System.Object)"]

Implements

[IGraphChangedListener.EdgePropertyRemoved\(IEdge, String, Object\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.EdgePropertyRemovedEventRecorded Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyRemovedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long EdgePropertyRemovedEventRecorded()
```

VB

```
Public Function EdgePropertyRemovedEventRecorded As Long
```

C++

```
public:  
long long EdgePropertyRemovedEventRecorded()
```

F#

```
member EdgePropertyRemovedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgePropertyRemovedEventRecorded"]

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.EdgeRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EdgeRemoved(  
    IEdge edge,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub EdgeRemoved (  
    edge As IEdge,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
virtual void EdgeRemoved(  
    IEdge^ edge,  
    IDictionary<String^, Object^>^ props  
) sealed
```

F#

```
abstract EdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit  
override EdgeRemoved :  
    edge : IEdge *  
    props : IDictionary<string, Object> -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

**[Missing <param name="props" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.EdgeRemoved(VelocityGraph.Frontenac.Blueprints.IEdge,System.Collections.Generic.IDictionary{System.String,System.Object})"]**

Implements

[IGraphChangedListener.EdgeRemoved\(IEdge, IDictionary{String, Object}\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.EdgeRemovedEventRecorded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgeRemovedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long EdgeRemovedEventRecorded ()
```

VB

```
Public Function EdgeRemovedEventRecorded As Long
```

C++

```
public:  
long long EdgeRemovedEventRecorded ()
```

F#

```
member EdgeRemovedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.EdgeRemovedEventRecorded"]

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.GetOrder Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.GetOrder"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public List<string> GetOrder ()
```

VB

```
Public Function GetOrder As List(Of String)
```

C++

```
public:  
List<String^>^ GetOrder ()
```

F#

```
member GetOrder : unit -> List<string>
```

Return Value

Type: [List\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.GetOrder"]

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.Reset Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.Reset"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Reset ()
```

VB

```
Public Sub Reset
```

C++

```
public:  
void Reset ()
```

F#

```
member Reset : unit -> unit
```

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexAdded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexAdded(  
    IVertex vertex  
)
```

VB

```
Public Sub VertexAdded (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void VertexAdded(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract VertexAdded :  
    vertex : IVertex -> unit  
override VertexAdded :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexAdded(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraphChangedListener.VertexAdded\(IVertex\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexPropertyChanged Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexPropertyChanged(  
    IVertex vertex,  
    string s,  
    Object o,  
    Object n  
)
```

VB

```
Public Sub VertexPropertyChanged (  
    vertex As IVertex,  
    s As String,  
    o As Object,  
    n As Object  
)
```

C++

```
public:  
virtual void VertexPropertyChanged(  
    IVertex^ vertex,  
    String^ s,  
    Object^ o,  
    Object^ n  
) sealed
```

F#

```
abstract VertexPropertyChanged :  
    vertex : IVertex *  
    s : string *  
    o : Object *  
    n : Object -> unit  
override VertexPropertyChanged :  
    vertex : IVertex *  
    s : string *  
    o : Object *  
    n : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

s

Type: [System.String](#)

[Missing <param name="s"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

o

Type: [System.Object](#)

[Missing <param name="o"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

n

Type: [System.Object](#)

[Missing <param name="n"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChanged(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Implements

[IGraphChangeListener.VertexPropertyChanged\(IVertex, String, Object, Object\)](#)

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangeListener.VertexPropertyChangedEventRecorded Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChangedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long VertexPropertyChangedEventRecorded()
```

VB

```
Public Function VertexPropertyChangedEventRecorded As Long
```

C++

```
public:  
long long VertexPropertyChangedEventRecorded()
```

F#

```
member VertexPropertyChangedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangeListener.VertexPropertyChangedEventRecorded"]

See Also

[StubGraphChangeListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexPropertyRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexPropertyRemoved(  
    IVertex vertex,  
    string s,  
    Object o  
)
```

VB

```
Public Sub VertexPropertyRemoved (  
    vertex As IVertex,  
    s As String,  
    o As Object  
)
```

C++

```
public:  
virtual void VertexPropertyRemoved(  
    IVertex^ vertex,  
    String^ s,  
    Object^ o  
) sealed
```

F#

```
abstract VertexPropertyRemoved :  
    vertex : IVertex *  
    s : string *  
    o : Object -> unit  
override VertexPropertyRemoved :  
    vertex : IVertex *  
    s : string *  
    o : Object -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

s

Type: [System.String](#)

[Missing <param name="s"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

o

Type: [System.Object](#)

[Missing <param name="o"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

Implements

[IGraphChangedListener.VertexPropertyRemoved\(IVertex, String, Object\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexPropertyRemovedEventRecorded Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemovedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long VertexPropertyRemovedEventRecorded()
```

VB

```
Public Function VertexPropertyRemovedEventRecorded As Long
```

C++

```
public:  
long long VertexPropertyRemovedEventRecorded()
```

F#

```
member VertexPropertyRemovedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexPropertyRemovedEventRecorded"]

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexRemoved Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void VertexRemoved(  
    IVertex vertex,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub VertexRemoved (  
    vertex As IVertex,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
virtual void VertexRemoved(  
    IVertex^ vertex,  
    IDictionary<String^, Object^>^ props  
) sealed
```

F#

```
abstract VertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit  
override VertexRemoved :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

**[Missing <param name="props" /> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexRemoved(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]**

Implements

[IGraphChangedListener.VertexRemoved\(IVertex, IDictionary\(String, Object\)\)](#)

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

StubGraphChangedListener.VertexRemovedEventRecorded Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexRemovedEventRecorded"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long VertexRemovedEventRecorded ()
```

VB

```
Public Function VertexRemovedEventRecorded As Long
```

C++

```
public:  
long long VertexRemovedEventRecorded ()
```

F#

```
member VertexRemovedEventRecorded : unit -> int64
```

Return Value

Type: [Int64](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.StubGraphChangedListener.VertexRemovedEventRecorded"]

See Also

[StubGraphChangedListener Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexAddedEvent Class

Event that fires when a vertex is added to a graph.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexAddedEvent

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class VertexAddedEvent : IEvent
```

VB

```
Public Class VertexAddedEvent  
    Implements IEvent
```

C++


```
public ref class VertexAddedEvent : IEvent
```

F#


```
type VertexAddedEvent =  
    class  
        interface IEvent  
    end
```

The **VertexAddedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------------------|-----------------------------------------------------------------|
|  | VertexAddedEvent | Initializes a new instance of the VertexAddedEvent class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexAddedEvent Constructor

Initializes a new instance of the [VertexAddedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexAddedEvent (  
    IVertex vertex  
)
```

VB

```
Public Sub New (  
    vertex As IVertex  
)
```

C++

```
public:  
VertexAddedEvent (  
    IVertex^ vertex  
)
```

F#

```
new :  
    vertex : IVertex -> VertexAddedEvent
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexAddedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also


[VertexAddedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexAddedEvent.VertexAddedEvent Methods

The [VertexAddedEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VertexAddedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexAddedEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexAddedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
virtual void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
) sealed
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
override FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexAddedEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerable\(IGraphChangedListener\)\)](#)

See Also

[VertexAddedEvent Class](#)

VertexPropertyChangedEvent Class

Event that fires when a property changes on a vertex.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEvent](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class VertexPropertyChangedEvent : VertexPropertyEvent
```

VB

```
Public Class VertexPropertyChangedEvent  
    Inherits VertexPropertyEvent
```

C++


```
public ref class VertexPropertyChangedEvent : public VertexPropertyEvent
```

F#

```
type VertexPropertyChangedEvent =  
    class  
        inherit VertexPropertyEvent  
    end
```

The **VertexPropertyChangedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------|
|  | VertexPropertyChangedEvent | Initializes a new instance of the VertexPropertyChangedEvent class |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyChangedEvent Constructor

Initializes a new instance of the [VertexPropertyChangedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexPropertyChangedEvent (  
    IVertex vertex,  
    string key,  
    Object oldValue,  
    Object newValue  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    key As String,  
    oldValue As Object,  
    newValue As Object  
)
```

C++

```
public:  
VertexPropertyChangedEvent (  
    IVertex^ vertex,  
    String^ key,  
    Object^ oldValue,  
    Object^ newValue  
)
```

F#

```
new :  
    vertex : IVertex *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> VertexPropertyChangedEvent
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

**[Missing <param name="key"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent.#
ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]**

oldValue

Type: [System.Object](#)

**[Missing <param name="oldValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent.#
ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]**

newValue

Type: [System.Object](#)

**[Missing <param name="newValue"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent.#
ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]**

See Also

[VertexPropertyChangedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyEvent Class

Base class for property changed events.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEvent
[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyChangedEvent](#)
[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyRemovedEvent](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class VertexPropertyEvent : IEvent
```

VB

```
Public MustInherit Class VertexPropertyEvent  
    Implements IEvent
```

C++


```
public ref class VertexPropertyEvent abstract : IEvent
```

F#

```
[<AbstractClassAttribute>]  
type VertexPropertyEvent =  
    class  
        interface IEvent  
    end
```

The **VertexPropertyEvent** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyEvent.VertexPropertyEvent Methods

The [VertexPropertyEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VertexPropertyEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (
    IEnumerable<IGraphChangedListener> eventListeners
)
```

VB

```
Public Sub FireEvent (
    eventListeners As IEnumerable(Of IGraphChangedListener)
)
```

C++

```
public:
virtual void FireEvent (
    IEnumerable<IGraphChangedListener^>^ eventListeners
) sealed
```

F#

```
abstract FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
override FireEvent :
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEvent.FireEvent(System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerator\(IGraphChangedListener\)\)](#)

See Also

[VertexPropertyEvent Class](#)

VertexPropertyEventContract Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class VertexPropertyEventContract
```

VB

```
Public NotInheritable Class VertexPropertyEventContract
```

C++

```
public ref class VertexPropertyEventContract abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type VertexPropertyEventContract = class end
```

The **VertexPropertyEventContract** type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------|
|  | ValidateFire | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyEventContract.VertexPropertyEventContract Methods

The [VertexPropertyEventContract](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------|
|  | ValidateFire | |

See Also

[VertexPropertyEventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyEventContract.ValidateFire Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static void ValidateFire(  
    IGraphChangedListener listener,  
    IVertex vertex,  
    string key,  
    Object oldValue,  
    Object newValue  
)
```

VB

```
Public Shared Sub ValidateFire (  
    listener As IGraphChangedListener,  
    vertex As IVertex,  
    key As String,  
    oldValue As Object,  
    newValue As Object  
)
```

C++

```
public:  
static void ValidateFire(  
    IGraphChangedListener^ listener,  
    IVertex^ vertex,  
    String^ key,  
    Object^ oldValue,  
    Object^ newValue  
)
```

F#

```
static member ValidateFire :  
    listener : IGraphChangedListener *  
    vertex : IVertex *  
    key : string *  
    oldValue : Object *  
    newValue : Object -> unit
```

Parameters

listener

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener](#)

[Missing <param name="listener"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

oldValue

Type: [System.Object](#)

[Missing <param name="oldValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

newValue

Type: [System.Object](#)

[Missing <param name="newValue"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEventContract.ValidateFire(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener, VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object,System.Object)"]

See Also

[VertexPropertyEventContract Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyRemovedEvent Class

Event fired when a vertex property is removed.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyEvent](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyRemovedEvent](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class VertexPropertyRemovedEvent : VertexPropertyEvent
```

VB

```
Public Class VertexPropertyRemovedEvent  
    Inherits VertexPropertyEvent
```

C++


```
public ref class VertexPropertyRemovedEvent : public VertexPropertyEvent
```

F#

```
type VertexPropertyRemovedEvent =  
    class  
        inherit VertexPropertyEvent  
    end
```

The **VertexPropertyRemovedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------|
|  | VertexPropertyRemovedEvent | Initializes a new instance of the VertexPropertyRemovedEvent class |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexPropertyRemovedEvent Constructor

Initializes a new instance of the [VertexPropertyRemovedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexPropertyRemovedEvent (  
    IVertex vertex,  
    string key,  
    Object removedValue  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    key As String,  
    removedValue As Object  
)
```

C++

```
public:  
VertexPropertyRemovedEvent (  
    IVertex^ vertex,  
    String^ key,  
    Object^ removedValue  
)
```

F#

```
new :  
    vertex : IVertex *  
    key : string *  
    removedValue : Object -> VertexPropertyRemovedEvent
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

removedValue

Type: [System.Object](#)

[Missing <param name="removedValue"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexPropertyRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.String,System.Object)"]

See Also

[VertexPropertyRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexRemovedEvent Class

Event fired when a vertex is removed.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexRemovedEvent

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class VertexRemovedEvent : IEvent
```

VB

```
Public Class VertexRemovedEvent  
    Implements IEvent
```

C++


```
public ref class VertexRemovedEvent : IEvent
```

F#


```
type VertexRemovedEvent =  
    class  
        interface IEvent  
    end
```

The **VertexRemovedEvent** type exposes the following members.

Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------|
|  | VertexRemovedEvent | Initializes a new instance of the VertexRemovedEvent class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexRemovedEvent Constructor

Initializes a new instance of the [VertexRemovedEvent](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public VertexRemovedEvent (  
    IVertex vertex,  
    IDictionary<string, Object> props  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    props As IDictionary(Of String, Object)  
)
```

C++

```
public:  
VertexRemovedEvent (  
    IVertex^ vertex,  
    IDictionary<String^, Object^>^ props  
)
```

F#

```
new :  
    vertex : IVertex *  
    props : IDictionary<string, Object> -> VertexRemovedEvent
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

props

Type: [System.Collections.Generic.IDictionary\(String, Object\)](#)

[Missing <param name="props"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexRemovedEvent.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,System.Collections.Generic.IDictionary{System.String,System.Object})"]

VelocityDB Class Library

See Also


[VertexRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexRemovedEvent.VertexRemovedEvent Methods

The [VertexRemovedEvent](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | FireEvent | |

See Also

[VertexRemovedEvent Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener Namespace](#)

VertexRemovedEvent.FireEvent Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexRemovedEvent.FireEvent (System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void FireEvent (  
    IEnumerable<IGraphChangedListener> eventListeners  
)
```

VB

```
Public Sub FireEvent (  
    eventListeners As IEnumerable(Of IGraphChangedListener)  
)
```

C++

```
public:  
virtual void FireEvent (  
    IEnumerable<IGraphChangedListener^>^ eventListeners  
) sealed
```

F#

```
abstract FireEvent :  
    eventListeners : IEnumerable<IGraphChangedListener> -> unit  
override FireEvent :  
    eventListeners : IEnumerable<IGraphChangedListener> -> unit
```

Parameters

eventListeners

Type: [System.Collections.Generic.IEnumerator\(IGraphChangedListener\)](#)

[Missing <param name="eventListeners"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.VertexRemovedEvent.FireEvent (System.Collections.Generic.IEnumerator{VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Event.Listener.IGraphChangedListener})"]

Implements

[IEvent.FireEvent\(IEnumerator\(IGraphChangedListener\)\)](#)







See Also

[VertexRemovedEvent Class](#)


VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id"]

Classes

| Class | Description |
|-----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  IdEdge | |
|  IdEdgeIndex | |
|  IdElement | |
|  IdGraph | A Graph implementation which wraps another Graph implementation, enabling custom element IDs even for those graphs which don't otherwise support them. The base Graph must be an instance of KeyIndexableGraph. It <i>may</i> be an instance of IIndexableGraph, in which case its indices will be wrapped appropriately. It <i>may</i> be an instance of TransactionalGraph, in which case transaction operations will be passed through. For those graphs which support vertex indices but not edge indices (or vice versa), you may configure idInnerTinkerGraph to use custom IDs only for vertices or only for edges. |
|  IdVertex | |
|  IdVertexIndex | |

Interfaces

| Interface | Description |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
|  IdGraph.IIdFactory | A factory for IDs of newly-created vertices and edges (where an ID is not otherwise specified). |

IdEdge Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class IdEdge : IdElement, IEdge,
    IElement, IDictionary<string, Object>, ICollection<KeyValuePair<string,
Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection
```

VB

```
Public Class IdEdge
    Inherits IdElement
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class IdEdge : public IdElement,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```


F#

```
type IdEdge =
    class
        inherit IdElement
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```
interface ICollection
end
```

The **IdEdge** type exposes the following members.




Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------------------------------------------------|
|  | IdEdge | Initializes a new instance of the IdEdge class |







Properties





| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------------|
|  | GetBaseEdge | |
|  | GetVertex | |
|  | ToString | (Overrides IdElement.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge Constructor

Initializes a new instance of the [IdEdge](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdEdge (
    IEdge baseEdge,
    IdGraph idInnerTinkerGraph
)
```

VB

```
Public Sub New (
    baseEdge As IEdge,
    idInnerTinkerGraph As IdGraph
)
```

C++

```
public:
    IdEdge (
        IEdge^ baseEdge,
        IdGraph^ idInnerTinkerGraph
    )
```

F#

```
new :
    baseEdge : IEdge *
    idInnerTinkerGraph : IdGraph -> IdEdge
```

Parameters

baseEdge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="baseEdge"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

idInnerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph](#)

[Missing <param name="idInnerTinkerGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

See Also

[IdEdge Class](#)


VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge.IdEdge Properties

The [IdEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

See Also

[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge.Label Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Label { get; }
```

VB

```
Public ReadOnly Property Label As String  
    Get
```

C++

```
public:  
virtual property String^ Label {  
    String^ get () sealed;  
}
```

F#

```
abstract Label : string with get  
override Label : string with get
```

Property Value

Type: [String](#)

Implements

[IEdge.Label](#)

See Also




[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)











IdEdge.IdEdge Methods

The [IdEdge](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------------|
|  | GetBaseEdge | |
|  | GetVertex | |
|  | ToString | (Overrides IdElement.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

| | | |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|--|--|--------------------------------------------------------------------------------------------------------------------------------------------|

See Also

[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge.GetBaseEdge Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.GetBaseEdge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetBaseEdge ()
```

VB

```
Public Function GetBaseEdge As IEdge
```

C++

```
public:  
IEdge^ GetBaseEdge ()
```

F#

```
member GetBaseEdge : unit -> IEdge
```

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.GetBaseEdge"]

See Also

[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Public Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Direction direction  
) sealed
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex  
override GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IEdge.GetVertex\(Direction\)](#)

See Also

[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdge.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge.ToString"]

See Also

[IdEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class IdEdgeIndex : IIndex
```

VB

```
Public Class IdEdgeIndex  
    Implements IIndex
```

C++


```
public ref class IdEdgeIndex : IIndex
```

F#



```
type IdEdgeIndex =  
    class  
        interface IIndex  
    end
```

The **IdEdgeIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------|
|  | IdEdgeIndex | Initializes a new instance of the IdEdgeIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex Constructor

Initializes a new instance of the [IdEdgeIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdEdgeIndex(  
    IIndex baseIndex,  
    IdGraph idGraph  
)
```

VB

```
Public Sub New (  
    baseIndex As IIndex,  
    idGraph As IdGraph  
)
```

C++

```
public:  
IdEdgeIndex(  
    IIndex^ baseIndex,  
    IdGraph^ idGraph  
)
```

F#

```
new :  
    baseIndex : IIndex *  
    idGraph : IdGraph -> IdEdgeIndex
```

Parameters

baseIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="baseIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

idGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph](#)

[Missing <param name="idGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

See Also

[IdEdgeIndex Class](#)



VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.IdEdgeIndex Properties

The [IdEdgeIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Name Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Type Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.IdEdgeIndex Methods

The [IdEdgeIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Count Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Count(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Count(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Count(System.String,System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Count(System.String,System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Query(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Query(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Query(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Query(System.String,System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdEdgeIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdgeIndex.ToString"]

See Also

[IdEdgeIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdEdge](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class IdElement : DictionaryElement
```

VB

```
Public MustInherit Class IdElement
    Inherits DictionaryElement
```

C++


```
public ref class IdElement abstract : public DictionaryElement
```

F#


```
[<AbstractClassAttribute>]
type IdElement =
    class
        inherit DictionaryElement
    end
```

The **IdElement** type exposes the following members.







Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id .) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  Remove | (Overrides DictionaryElement.Remove().) |
|  RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  ToString | (Overrides Object.ToString().) |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.IdElement Properties

The [IdElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.Id Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also









[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.IdElement Methods

The [IdElement](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys() .) |
|  | Remove | (Overrides DictionaryElement.Remove() .) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String) .) |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.Equals Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.Equals(System.Object)"]

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.GetHashCode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetHashCode"]

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetProperty(System.String)"]

Implements

[IElement.GetProperty\(String\)](#)

VelocityDB Class Library

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.GetPropertyKeys"]

Implements

[IElement.GetPropertyKeys\(\)](#)

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.Remove Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.RemoveProperty(System.String)"
]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.RemoveProperty(System.String)"
]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.RemoveProperty(System.String)"
]

VelocityDB Class Library

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.SetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.SetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.SetProperty(System.String,System.Object)"]

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdElement.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement.ToString"]

See Also

[IdElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph Class

A Graph implementation which wraps another Graph implementation, enabling custom element IDs even for those graphs which don't otherwise support them. The base Graph must be an instance of `IKeyIndexableGraph`. It *may* be an instance of `IIndexableGraph`, in which case its indices will be wrapped appropriately. It *may* be an instance of `ITransactionalGraph`, in which case transaction operations will be passed through. For those graphs which support vertex indices but not edge indices (or vice versa), you may configure `idInnerTinkerGraph` to use custom IDs only for vertices or only for edges.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class IdGraph : IKeyIndexableGraph, IGraph,
    IWrapperGraph, IIndexableGraph, ITransactionalGraph
```

VB

```
Public Class IdGraph
    Implements IKeyIndexableGraph, IGraph, IWrapperGraph, IIndexableGraph,
    ITransactionalGraph
```

C++



```
public ref class IdGraph : IKeyIndexableGraph,
    IGraph, IWrapperGraph, IIndexableGraph, ITransactionalGraph
```

F#




```
type IdGraph =
    class
        interface IKeyIndexableGraph
        interface IGraph
        interface IWrapperGraph
        interface IIndexableGraph
        interface ITransactionalGraph
    end
```

The **IdGraph** type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | IdGraph(IKeyIndexableGraph) | Adds custom ID functionality to the given graph, supporting both custom vertex IDs and custom edge IDs. |
|  | IdGraph(IKeyIndexableGraph, Boolean, Boolean) | Adds custom ID functionality to the given graph, supporting either custom vertex IDs, custom edge IDs, or both. |

Properties



| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------|
|  | EdgeIdFactory | When edges are created using null IDs, the actual IDs are chosen based on this factory. |
|  | Features | |
|  | VertexIdFactory | When edges are created using null IDs, the actual IDs are chosen based on this factory. |

Methods























| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | CreateIndex | |
|  | CreateKeyIndex | |
|  | DropIndex | |
|  | DropKeyIndex | |
|  | EnforceUniques | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetIndex | |
|  | GetIndexedKeys | |
|  | GetIndices | |
|  | GetSupportEdgeIds | |
|  | GetSupportVertexIds | |
|  | GetVertex | |
|  | GetVertices() | |

| | |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
|  GetVertices(String, Object) | |
|  Query | |
|  RemoveEdge | |
|  RemoveVertex | |
|  Rollback | |
|  Shutdown | |
|  ToString | (Overrides Object.ToString() .) |

Fields

| | Name | Description |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|-------------|
|   | Id | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|   | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|   | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|   | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|   | CreateTinkerGraph | (Defined by GraphHelpers .) |
|   | GraphString | (Defined by StringFactory .) |
|   | LoadGml | (Defined by GraphHelpers .) |
|   | LoadGraphml | (Defined by GraphHelpers .) |
|   | LoadGraphson | (Defined by GraphHelpers .) |
|   | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|   | SaveDotNet | (Defined by GraphHelpers .) |

VelocityDB Class Library



| | |
|----------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | IdGraph(IKeyIndexableGraph) | Adds custom ID functionality to the given graph, supporting both custom vertex IDs and custom edge IDs. |
|  | IdGraph(IKeyIndexableGraph, Boolean, Boolean) | Adds custom ID functionality to the given graph, supporting either custom vertex IDs, custom edge IDs, or both. |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph Constructor (IKeyIndexableGraph)

Adds custom ID functionality to the given graph, supporting both custom vertex IDs and custom edge IDs.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdGraph(  
    IKeyIndexableGraph baseGraph  
)
```

VB

```
Public Sub New (  
    baseGraph As IKeyIndexableGraph  
)
```

C++

```
public:  
IdGraph(  
    IKeyIndexableGraph^ baseGraph  
)
```

F#

```
new :  
    baseGraph : IKeyIndexableGraph -> IdGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IKeyIndexableGraph](#)

the base graph which does not necessarily support custom IDs

See Also

[IdGraph Class](#)

[IdGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph Constructor (IKeyIndexableGraph, Boolean, Boolean)

Adds custom ID functionality to the given graph, supporting either custom vertex IDs, custom edge IDs, or both.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdGraph(  
    IKeyIndexableGraph baseGraph,  
    bool supportVertexIds,  
    bool supportEdgeIds  
)
```

VB

```
Public Sub New (  
    baseGraph As IKeyIndexableGraph,  
    supportVertexIds As Boolean,  
    supportEdgeIds As Boolean  
)
```

C++

```
public:  
IdGraph(  
    IKeyIndexableGraph^ baseGraph,  
    bool supportVertexIds,  
    bool supportEdgeIds  
)
```

F#

```
new :  
    baseGraph : IKeyIndexableGraph *  
    supportVertexIds : bool *  
    supportEdgeIds : bool -> IdGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IKeyIndexableGraph](#)
the base graph which does not necessarily support custom IDs

supportVertexIds

Type: [System.Boolean](#)

whether to support custom vertex IDs

supportEdgeIds

Type: [System.Boolean](#)

whether to support custom edge IDs

See Also

[IdGraph Class](#)




[IdGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.IdGraph Properties

The [IdGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------------------|
|  | EdgeIdFactory | When edges are created using null IDs, the actual IDs are chosen based on this factory. |
|  | Features | |
|  | VertexIdFactory | When edges are created using null IDs, the actual IDs are chosen based on this factory. |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.EdgeIdFactory Property

When edges are created using null IDs, the actual IDs are chosen based on this factory.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdGraph.IIdFactory EdgeIdFactory { get; set; }
```

VB

```
Public Property EdgeIdFactory As IdGraph.IIdFactory  
    Get  
    Set
```

C++

```
public:  
property IdGraph.IIdFactory^ EdgeIdFactory {  
    IdGraph.IIdFactory^ get ();  
    void set (IdGraph.IIdFactory^ value);  
}
```

F#

```
member EdgeIdFactory : IdGraph.IIdFactory with get, set
```

Property Value

Type: [IdGraph.IIdFactory](#)

the factory for new edge IDs.

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Features Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.VertexIdFactory Property

When edges are created using null IDs, the actual IDs are chosen based on this factory.

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdGraph.IIdFactory VertexIdFactory { get; set; }
```

VB

```
Public Property VertexIdFactory As IdGraph.IIdFactory  
    Get  
    Set
```

C++

```
public:  
property IdGraph.IIdFactory^ VertexIdFactory {  
    IdGraph.IIdFactory^ get ();  
    void set (IdGraph.IIdFactory^ value);  
}
```

F#

```
member VertexIdFactory : IdGraph.IIdFactory with get, set
```

Property Value

Type: [IdGraph.IIdFactory](#)

the factory for new vertex IDs.

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)


IdGraph.IdGraph Methods














The [IdGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | Commit | |
|  | CreateIndex | |
|  | CreateKeyIndex | |
|  | DropIndex | |
|  | DropKeyIndex | |
|  | EnforceUniqueIds | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetIndex | |
|  | GetIndexedKeys | |
|  | GetIndices | |
|  | GetSupportEdgeIds | |
|  | GetSupportVertexIds | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Rollback | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  CreateTinkerGraph | (Defined by GraphHelpers .) |
|  GraphString | (Defined by StringFactory .) |
|  LoadGml | (Defined by GraphHelpers .) |
|  LoadGraphml | (Defined by GraphHelpers .) |
|  LoadGraphson | (Defined by GraphHelpers .) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  SaveDotNet | (Defined by GraphHelpers .) |
|  SaveGml | (Defined by GraphHelpers .) |
|  SaveGraphml | (Defined by GraphHelpers .) |
|  SaveGraphson | (Defined by GraphHelpers .) |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.AddVertex Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.AddVertex(System.Object)"]

Implements

[IGraph.AddVertex\(Object\)](#)

VelocityDB Class Library

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Commit Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Commit"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Commit ()
```

VB

```
Public Sub Commit
```

C++

```
public:  
virtual void Commit () sealed
```

F#

```
abstract Commit : unit -> unit  
override Commit : unit -> unit
```

Implements

[ITransactionalGraph.Commit\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.CreateIndex Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateIndex(System.String,System.
Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateIndex(System.String,System.
Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateIndex(System.String,System.
Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.CreateKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CreateKeyIndex(  
    string key,  
    Type elementClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Sub CreateKeyIndex (  
    key As String,  
    elementClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
public:  
virtual void CreateKeyIndex(  
    String^ key,  
    Type^ elementClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit  
override CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IKeyIndexableGraph.CreateKeyIndex\(String, Type, Parameter\[\]\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.DropIndex Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.DropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropIndex(  
    string indexName  
)
```

VB

```
Public Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
) sealed
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.DropIndex(System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.DropKeyIndex Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.DropKeyIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropKeyIndex(  
    string key,  
    Type elementClass  
)
```

VB

```
Public Sub DropKeyIndex (  
    key As String,  
    elementClass As Type  
)
```

C++

```
public:  
virtual void DropKeyIndex(  
    String^ key,  
    Type^ elementClass  
) sealed
```

F#

```
abstract DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit  
override DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.DropKeyIndex(System.String,System.Type)"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.DropKeyIndex(System.String,System.Type)"]

Implements

[IKeyIndexableGraph.DropKeyIndex\(String, Type\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.EnforceUniqueIds Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.EnforceUniqueIds(System.Boolean)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void EnforceUniqueIds (  
    bool enforceUniqueIds  
)
```

VB

```
Public Sub EnforceUniqueIds (  
    enforceUniqueIds As Boolean  
)
```

C++

```
public:  
void EnforceUniqueIds (  
    bool enforceUniqueIds  
)
```

F#

```
member EnforceUniqueIds :  
    enforceUniqueIds : bool -> unit
```

Parameters

enforceUniqueIds

Type: [System.Boolean](#)

[Missing <param name="enforceUniqueIds"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.EnforceUniqueIds(System.Boolean)"]

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetBaseGraph Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdge(System.Object)"]

Implements

[IGraph.GetEdge\(Object\)](#)

VelocityDB Class Library

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable (Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[IdGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[IdGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
) sealed
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndex(System.String,System.Type)"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndex(System.String,System.Type)"]

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetIndexedKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndexedKeys(System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> GetIndexedKeys (  
    Type elementClass  
)
```

VB

```
Public Function GetIndexedKeys (  
    elementClass As Type  
) As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetIndexedKeys (  
    Type^ elementClass  
) sealed
```

F#

```
abstract GetIndexedKeys :  
    elementClass : Type -> IEnumerable<string>  
override GetIndexedKeys :  
    elementClass : Type -> IEnumerable<string>
```

Parameters

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndexedKeys(System.Type)"]

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndexedKeys(System.Type)"]

Implements

[IKeyIndexableGraph.GetIndexedKeys\(Type\)](#)

VelocityDB Class Library

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetIndices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Function GetIndices As IEnumerable(Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndex^>^ GetIndices () sealed
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetIndices"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetSupportEdgeIds Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetSupportEdgeIds"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GetSupportEdgeIds ()
```

VB

```
Public Function GetSupportEdgeIds As Boolean
```

C++

```
public:  
bool GetSupportEdgeIds ()
```

F#

```
member GetSupportEdgeIds : unit -> bool
```

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetSupportEdgeIds"]

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetSupportVertexIds Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetSupportVertexIds"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool GetSupportVertexIds ()
```

VB

```
Public Function GetSupportVertexIds As Boolean
```

C++

```
public:  
bool GetSupportVertexIds ()
```

F#

```
member GetSupportVertexIds : unit -> bool
```

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetSupportVertexIds"]

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertex(System.Object)"]

Implements

[IGraph.GetVertex\(Object\)](#)

VelocityDB Class Library

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetVertices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[IdGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetVertices (  
    key As String,  
    value As Object  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>  
override GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[IdGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (  
    IEdge edge  
)
```

VB

```
Public Sub RemoveEdge (  
    edge As IEdge  
)
```

C++

```
public:  
virtual void RemoveEdge (  
    IEdge^ edge  
) sealed
```

F#

```
abstract RemoveEdge :  
    edge : IEdge -> unit  
override RemoveEdge :  
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Rollback Method

[Missing <summary> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Rollback"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Rollback()
```

VB

```
Public Sub Rollback
```

C++

```
public:  
virtual void Rollback() sealed
```

F#

```
abstract Rollback : unit -> unit  
override Rollback : unit -> unit
```

Implements

[ITransactionalGraph.Rollback\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public virtual void Shutdown ()
```

VB

```
Public Overridable Sub Shutdown
```

C++

```
public:  
virtual void Shutdown ()
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.ToString"]

See Also


[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.IdGraph Fields

The [IdGraph](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|-------------|
|  | Id | |

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.Id Field

[Missing <summary> documentation for "F:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string Id = "__id"
```

VB

```
Public Const Id As String = "__id"
```

C++

```
public:  
literal String^ Id = "__id"
```

F#

```
static val mutable Id: string
```

Field Value

Type: [String](#)

See Also

[IdGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.IIdFactory Interface

A factory for IDs of newly-created vertices and edges (where an ID is not otherwise specified).

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public interface IIdFactory
```

VB

```
Public Interface IIdFactory
```

C++


```
public interface class IIdFactory
```

F#

```
type IIdFactory = interface end
```

The IdGraph.IIdFactory type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Createld | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdFactory.IdFactory Methods

The [IdGraph.IdFactory](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | CreateId | |

See Also

[IdGraph.IdFactory Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdGraph.IIdFactory.CreateId Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IIdFactory.CreateId"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
Object CreateId()
```

VB

```
Function CreateId As Object
```

C++

```
Object^ CreateId()
```

F#

```
abstract CreateId : unit -> Object
```

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IIdFactory.CreateId"]

See Also

[IdGraph.IIdFactory Interface](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class IdVertex : IdElement, IVertex,
    IElement, IDictionary<string, Object>, ICollection<KeyValuePair<string,
Object>>,
    IEnumerable<KeyValuePair<string, Object>>, IEnumerable,
    IDictionary, ICollection
```

VB

```
Public Class IdVertex
    Inherits IdElement
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class IdVertex : public IdElement,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type IdVertex =
    class
        inherit IdElement
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end







```

The **IdVertex** type exposes the following members.








Constructors





| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|---------------------------------------------------------|
|  | IdVertex | Initializes a new instance of the IdVertex class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------|
|  | AddEdge | |
|  | GetBaseVertex | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |
|  | ToString | (Overrides IdElement.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |
|  | VertexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex Constructor

Initializes a new instance of the [IdVertex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdVertex(  
    IVertex baseVertex,  
    IdGraph idInnerTinkerGraph  
)
```

VB

```
Public Sub New (  
    baseVertex As IVertex,  
    idInnerTinkerGraph As IdGraph  
)
```

C++

```
public:  
IdVertex(  
    IVertex^ baseVertex,  
    IdGraph^ idInnerTinkerGraph  
)
```

F#

```
new :  
    baseVertex : IVertex *  
    idInnerTinkerGraph : IdGraph -> IdVertex
```

Parameters

baseVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="baseVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

idInnerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph](#)

[Missing <param name="idInnerTinkerGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

See Also

[IdVertex Class](#)







VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)



IdVertex.IdVertex Methods

The [IdVertex](#) type exposes the following members.



Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------|
|  | AddEdge | |
|  | GetBaseVertex | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |
|  | ToString | (Overrides IdElement.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |

VelocityDB Class Library

| | |
|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |
|  VertexString | (Defined by StringFactory .) |

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    string label,
    IVertex vertex
)
```

VB

```
Public Function AddEdge (
    id As Object,
    label As String,
    vertex As IVertex
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    String^ label,
    IVertex^ vertex
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
override AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.GetBaseVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetBaseVertex"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetBaseVertex()
```

VB

```
Public Function GetBaseVertex As IVertex
```

C++

```
public:  
IVertex^ GetBaseVertex()
```

F#

```
member GetBaseVertex : unit -> IVertex
```

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetBaseVertex"]

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.GetVertices Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Public Function GetVertices (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    Direction direction,  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>  
override GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.Query"]

Implements

[IVertex.Query\(\)](#)

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertex.ToString"]

See Also

[IdVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class IdVertexIndex : IIndex
```

VB

```
Public Class IdVertexIndex  
    Implements IIndex
```

C++


```
public ref class IdVertexIndex : IIndex
```

F#



```
type IdVertexIndex =  
    class  
        interface IIndex  
    end
```

The **IdVertexIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | IdVertexIndex | Initializes a new instance of the IdVertexIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex Constructor

Initializes a new instance of the [IdVertexIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IdVertexIndex(  
    IIndex baseIndex,  
    IdGraph idGraph  
)
```

VB

```
Public Sub New (  
    baseIndex As IIndex,  
    idGraph As IdGraph  
)
```

C++

```
public:  
IdVertexIndex(  
    IIndex^ baseIndex,  
    IdGraph^ idGraph  
)
```

F#

```
new :  
    baseIndex : IIndex *  
    idGraph : IdGraph -> IdVertexIndex
```

Parameters

baseIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="baseIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

idGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph](#)

[Missing <param name="idGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdGraph)"]

See Also

[IdVertexIndex Class](#)



VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.IdVertexIndex Properties

The [IdVertexIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Name Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Type Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.IdVertexIndex Methods

The [IdVertexIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Count Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Count(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Count(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Count(System.String,System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Count(System.String,System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Query(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Query(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Query(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Query(System.String,System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[IdVertexIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

IdVertexIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id.IdVertexIndex.ToString"]

See Also

[IdVertexIndex Class](#)







[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Id Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace

[Missing <summary> documentation for

"N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------|-------------|
|  | PartitionEdge | |
|  | PartitionElement | |
|  | PartitionGraph | |
|  | PartitionIndex | |
|  | PartitionIndexableGraph | |
|  | PartitionVertex | |

PartitionEdge Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PartitionEdge : PartitionElement,
    IEdge, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class PartitionEdge
    Inherits PartitionElement
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
    KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class PartitionEdge : public PartitionElement,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type PartitionEdge =
    class
        inherit PartitionElement
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end


```

The **PartitionEdge** type exposes the following members.


Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | PartitionEdge | Initializes a new instance of the PartitionEdge class |








Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | EdgeString | (Defined by StringFactory.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are <code>equal()</code> . (Defined by ElementHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be <code>equals()</code> . (Defined by ElementHelpers.) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionEdge Constructor

Initializes a new instance of the [PartitionEdge](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionEdge(  
    IEdge edge,  
    PartitionGraph innerTinkerGraph  
)
```

VB

```
Public Sub New (  
    edge As IEdge,  
    innerTinkerGraph As PartitionGraph  
)
```

C++

```
public:  
PartitionEdge(  
    IEdge^ edge,  
    PartitionGraph^ innerTinkerGraph  
)
```

F#

```
new :  
    edge : IEdge *  
    innerTinkerGraph : PartitionGraph -> PartitionEdge
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"

innerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph](#)

[Missing <param name="innerTinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"

VelocityDB Class Library

See Also


[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionEdge.PartitionEdge Properties

The [PartitionEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

See Also

[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionEdge.Label Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Label { get; }
```

VB

```
Public ReadOnly Property Label As String  
    Get
```

C++

```
public:  
virtual property String^ Label {  
    String^ get () sealed;  
}
```

F#

```
abstract Label : string with get  
override Label : string with get
```

Property Value

Type: [String](#)

Implements

[IEdge.Label](#)

See Also

[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)











PartitionEdge.PartitionEdge Methods

The [PartitionEdge](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any Element.equals() implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. ElementPropertiesRule that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a HashMap is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

VelocityDB Class Library

See Also

[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionEdge.GetBaseEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.GetBaseEdge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetBaseEdge ()
```

VB

```
Public Function GetBaseEdge As IEdge
```

C++

```
public:  
IEdge^ GetBaseEdge ()
```

F#

```
member GetBaseEdge : unit -> IEdge
```

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.GetBaseEdge"]

See Also

[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionEdge.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Public Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Direction direction  
) sealed
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex  
override GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IEdge.GetVertex\(Direction\)](#)

See Also

[PartitionEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionEdge](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class PartitionElement : DictionaryElement
```

VB

```
Public MustInherit Class PartitionElement
    Inherits DictionaryElement
```

C++


```
public ref class PartitionElement abstract : public DictionaryElement
```

F#



```
[<AbstractClassAttribute>]
type PartitionElement =
    class
        inherit DictionaryElement
    end
```

The **PartitionElement** type exposes the following members.









Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object).) |
|  | GetBaseElement | |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  GetHashCode | (Overrides Object.GetHashCode().) |
|  GetPartition | |
|  GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  Remove | (Overrides DictionaryElement.Remove().) |
|  RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  SetPartition | |
|  SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  ToString | (Overrides Object.ToString().) |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.PartitionElement Properties

The [PartitionElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.Id Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also












[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.PartitionElement Methods

The [PartitionElement](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetBaseElement | |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |
|  | GetPartition | |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys() .) |
|  | Remove | (Overrides DictionaryElement.Remove() .) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String) .) |
|  | SetPartition | |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.Equals Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.Equals(System.Object)"]

See Also

[PartitionElement Class](#)

PartitionElement.GetBaseElement Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetBaseElement"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IElement GetBaseElement ()
```

VB

```
Public Function GetBaseElement As IElement
```

C++

```
public:  
IElement^ GetBaseElement ()
```

F#

```
member GetBaseElement : unit -> IElement
```

Return Value

Type: [IElement](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetBaseElement"]

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.GetHashCode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetHashCode"]

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.GetPartition Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetPartition"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string GetPartition()
```

VB

```
Public Function GetPartition As String
```

C++

```
public:  
String^ GetPartition()
```

F#

```
member GetPartition : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetPartition"]

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.GetProperty\(String\)](#)

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.GetPropertyKeys"]

Implements

[IElement.GetPropertyKeys\(\)](#)

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.RemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.RemoveProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.RemoveProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.SetPartition Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.SetPartition(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void SetPartition(  
    string partition  
)
```

VB

```
Public Sub SetPartition (  
    partition As String  
)
```

C++

```
public:  
void SetPartition(  
    String^ partition  
)
```

F#

```
member SetPartition :  
    partition : string -> unit
```

Parameters

partition

Type: [System.String](#)

[Missing <param name="partition"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.SetPartition(System.String)"]

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.SetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.SetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.SetProperty(System.String,System.Object)"]

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionElement.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement.ToString"]

See Also

[PartitionElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PartitionGraph : IGraph,
    IWrapperGraph
```

VB

```
Public Class PartitionGraph
    Implements IGraph, IWrapperGraph
```

C++



```
public ref class PartitionGraph : IGraph,
    IWrapperGraph
```

F#




```
type PartitionGraph =
    class
        interface IGraph
        interface IWrapperGraph
    end
```

The **PartitionGraph** type exposes the following members.


Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------|
|  | PartitionGraph(IGraph, String, String) | Initializes a new instance of the PartitionGraph class |
|  | PartitionGraph(IGraph, String, String, IEnumerable(String)) | Initializes a new instance of the PartitionGraph class |




Properties











| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Features | |
|  | PartitionKey | |
|  | WritePartition | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddReadPartition | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetReadPartitions | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | IsInPartition | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveReadPartition | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This |



| | | |
|------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | method only works for graphs where the user can control the element ids. (Defined by GraphHelpers.) |
|  | CreateTinkerGraph | (Defined by GraphHelpers.) |
|  | GraphString | (Defined by StringFactory.) |
|  | LoadGml | (Defined by GraphHelpers.) |
|  | LoadGraphml | (Defined by GraphHelpers.) |
|  | LoadGraphson | (Defined by GraphHelpers.) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|------------------------------------------------------------------------|
|  | PartitionGraph(IGraph, String, String) | Initializes a new instance of the PartitionGraph class |
|  | PartitionGraph(IGraph, String, String, IEnumerable(String)) | Initializes a new instance of the PartitionGraph class |

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph Constructor (IGraph, String, String)

Initializes a new instance of the [PartitionGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionGraph(  
    IGraph baseGraph,  
    string partitionKey,  
    string readWritePartition  
)
```

VB

```
Public Sub New (  
    baseGraph As IGraph,  
    partitionKey As String,  
    readWritePartition As String  
)
```

C++

```
public:  
PartitionGraph(  
    IGraph^ baseGraph,  
    String^ partitionKey,  
    String^ readWritePartition  
)
```

F#

```
new :  
    baseGraph : IGraph *  
    partitionKey : string *  
    readWritePartition : string -> PartitionGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="baseGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String)"]

partitionKey

Type: [System.String](#)

[Missing <param name="partitionKey"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String)"]

readWritePartition

Type: [System.String](#)

[Missing <param name="readWritePartition"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String)"]

See Also

[PartitionGraph Class](#)

[PartitionGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph Constructor (IGraph, String, String, IEnumerable(String))

Initializes a new instance of the [PartitionGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionGraph(  
    IGraph baseGraph,  
    string partitionKey,  
    string writePartition,  
    IEnumerable<string> readPartitions  
)
```

VB

```
Public Sub New (  
    baseGraph As IGraph,  
    partitionKey As String,  
    writePartition As String,  
    readPartitions As IEnumerable(Of String)  
)
```

C++

```
public:  
PartitionGraph(  
    IGraph^ baseGraph,  
    String^ partitionKey,  
    String^ writePartition,  
    IEnumerable<String^>^ readPartitions  
)
```

F#

```
new :  
    baseGraph : IGraph *  
    partitionKey : string *  
    writePartition : string *  
    readPartitions : IEnumerable<string> -> PartitionGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="baseGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

partitionKey

Type: [System.String](#)

[Missing <param name="partitionKey"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

writePartition

Type: [System.String](#)

[Missing <param name="writePartition"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

readPartitions

Type: [System.Collections.Generic.IEnumerable\(String\)](#)

[Missing <param name="readPartitions"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[PartitionGraph Class](#)




[PartitionGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.PartitionGraph Properties

The [PartitionGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | Features | |
|  | PartitionKey | |
|  | WritePartition | |

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.PartitionKey Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.PartitionKey"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string PartitionKey { get; set; }
```

VB

```
Public Property PartitionKey As String  
    Get  
    Set
```

C++

```
public:  
property String^ PartitionKey {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member PartitionKey : string with get, set
```

Property Value

Type: [String](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.WritePartition Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.WritePartition"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string WritePartition { get; set; }
```

VB

```
Public Property WritePartition As String  
    Get  
    Set
```

C++

```
public:  
property String^ WritePartition {  
    String^ get ();  
    void set (String^ value);  
}
```

F#

```
member WritePartition : string with get, set
```

Property Value

Type: [String](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)







PartitionGraph.PartitionGraph Methods








The [PartitionGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddReadPartition | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetReadPartitions | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | IsInPartition | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveReadPartition | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  LoadGraphml | (Defined by GraphHelpers.) |
|  LoadGraphson | (Defined by GraphHelpers.) |
|  ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  SaveDotNet | (Defined by GraphHelpers.) |
|  SaveGml | (Defined by GraphHelpers.) |
|  SaveGraphml | (Defined by GraphHelpers.) |
|  SaveGraphson | (Defined by GraphHelpers.) |

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.AddReadPartition Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddReadPartition(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void AddReadPartition(  
    string readPartition  
)
```

VB

```
Public Sub AddReadPartition (  
    readPartition As String  
)
```

C++

```
public:  
void AddReadPartition(  
    String^ readPartition  
)
```

F#

```
member AddReadPartition :  
    readPartition : string -> unit
```

Parameters

readPartition

Type: [System.String](#)

[Missing <param name="readPartition"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddReadPartition(System.String)"]

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.AddVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.AddVertex\(Object\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdge(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetEdge\(Object\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[PartitionGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    string key,
    Object value
)
```

VB

```
Public Function GetEdges (
    key As String,
    value As Object
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
override GetEdges :
    key : string *
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[PartitionGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetReadPartitions Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetReadPartitions"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> GetReadPartitions ()
```

VB

```
Public Function GetReadPartitions As IEnumerable(Of String)
```

C++

```
public:  
IEnumerable<String^>^ GetReadPartitions ()
```

F#

```
member GetReadPartitions : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetReadPartitions"]

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetVertex\(Object\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[PartitionGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    string key,
    Object value
)
```

VB

```
Public Function GetVertices (
    key As String,
    value As Object
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
override GetVertices :
    key : string *
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[PartitionGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.IsInPartition Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.IsInPartition(VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public bool IsInPartition(  
    IElement element  
)
```

VB

```
Public Function IsInPartition (  
    element As IElement  
) As Boolean
```

C++

```
public:  
bool IsInPartition(  
    IElement^ element  
)
```

F#

```
member IsInPartition :  
    element : IElement -> bool
```

Parameters

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.IsInPartition(VelocityGraph.Frontenac.Blueprints.IElement)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.IsInPartition(VelocityGraph.Frontenac.Blueprints.IElement)"]

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (
    IEdge edge
)
```

VB

```
Public Sub RemoveEdge (
    edge As IEdge
)
```

C++

```
public:
virtual void RemoveEdge (
    IEdge^ edge
) sealed
```

F#

```
abstract RemoveEdge :
    edge : IEdge -> unit
override RemoveEdge :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.RemoveReadPartition Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveReadPartition (System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveReadPartition(  
    string readPartition  
)
```

VB

```
Public Sub RemoveReadPartition (  
    readPartition As String  
)
```

C++

```
public:  
void RemoveReadPartition(  
    String^ readPartition  
)
```

F#

```
member RemoveReadPartition :  
    readPartition : string -> unit
```

Parameters

readPartition

Type: [System.String](#)

[Missing <param name="readPartition"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveReadPartition (System.String)"]

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph.ToString"]

See Also

[PartitionGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class PartitionIndex : IIndex
```


```
VB
Public Class PartitionIndex
    Implements IIndex
```

```
C++
public ref class PartitionIndex : IIndex
```



```
F#
type PartitionIndex =
    class
        interface IIndex
    end
```

The **PartitionIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | PartitionIndex | Initializes a new instance of the PartitionIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex Constructor

Initializes a new instance of the [PartitionIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionIndex(  
    IIndex rawIndex,  
    PartitionGraph graph  
)
```

VB

```
Public Sub New (  
    rawIndex As IIndex,  
    graph As PartitionGraph  
)
```

C++

```
public:  
PartitionIndex(  
    IIndex^ rawIndex,  
    PartitionGraph^ graph  
)
```

F#

```
new :  
    rawIndex : IIndex *  
    graph : PartitionGraph -> PartitionIndex
```

Parameters

rawIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="rawIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"]

graph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"]

VelocityDB Class Library

See Also



[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.PartitionIndex Properties

The [PartitionIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Name Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Type Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)


PartitionIndex.PartitionIndex Methods

The [PartitionIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Count Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Count(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Count(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Count(System.String,System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Count(System.String,System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Query(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Query(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Query(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Query(System.String,System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Remove(System.String, System.Object, VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Remove(System.String, System.Object, VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Remove(System.String
,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.Remove(System.String
,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndex.ToString"]

See Also

[PartitionIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PartitionIndexableGraph : PartitionGraph,
    IIndexableGraph, IGraph
```

VB

```
Public Class PartitionIndexableGraph
    Inherits PartitionGraph
    Implements IIndexableGraph, IGraph
```

C++



```
public ref class PartitionIndexableGraph : public PartitionGraph,
    IIndexableGraph, IGraph
```

F#





```
type PartitionIndexableGraph =
    class
        inherit PartitionGraph
        interface IIndexableGraph
        interface IGraph
    end
```

The **PartitionIndexableGraph** type exposes the following members.















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
|  | PartitionIndexableGraph(IIndexableGraph, String, String) | Initializes a new instance of the PartitionIndexableGraph class |
|  | PartitionIndexableGraph(IIndexableGraph, String, String, IEnumerable(String)) | Initializes a new instance of the PartitionIndexableGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |



VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph Constructor

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
|  | PartitionIndexableGraph(IIndexableGraph, String, String) | Initializes a new instance of the PartitionIndexableGraph class |
|  | PartitionIndexableGraph(IIndexableGraph, String, String, IEnumerable(String)) | Initializes a new instance of the PartitionIndexableGraph class |

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph Constructor (IIndexableGraph, String, String)

Initializes a new instance of the [PartitionIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionIndexableGraph(  
    IIndexableGraph baseIndexableGraph,  
    string writeGraphKey,  
    string readWriteGraph  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph,  
    writeGraphKey As String,  
    readWriteGraph As String  
)
```

C++

```
public:  
PartitionIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph,  
    String^ writeGraphKey,  
    String^ readWriteGraph  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph *  
    writeGraphKey : string *  
    readWriteGraph : string -> PartitionIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String)"]

writeGraphKey

Type: [System.String](#)

[Missing <param name="writeGraphKey"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String)"]

readWriteGraph

Type: [System.String](#)

[Missing <param name="readWriteGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String)"]

See Also

[PartitionIndexableGraph Class](#)

[PartitionIndexableGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph Constructor (IIndexableGraph, String, String, IEnumerable(String))

Initializes a new instance of the [PartitionIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionIndexableGraph(  
    IIndexableGraph baseIndexableGraph,  
    string writeGraphKey,  
    string writeGraph,  
    IEnumerable<string> readGraphs  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph,  
    writeGraphKey As String,  
    writeGraph As String,  
    readGraphs As IEnumerable(Of String)  
)
```

C++

```
public:  
PartitionIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph,  
    String^ writeGraphKey,  
    String^ writeGraph,  
    IEnumerable<String^>^ readGraphs  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph *  
    writeGraphKey : string *  
    writeGraph : string *  
    readGraphs : IEnumerable<string> -> PartitionIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

writeGraphKey

Type: [System.String](#)

[Missing <param name="writeGraphKey"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

writeGraph

Type: [System.String](#)

[Missing <param name="writeGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

readGraphs

Type: [System.Collections.Generic.IEnumerable{String}](#)

[Missing <param name="readGraphs"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph,System.String,System.String,System.Collections.Generic.IEnumerable{System.String})"]

See Also

[PartitionIndexableGraph Class](#)





[PartitionIndexableGraph Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)














PartitionIndexableGraph.PartitionIndexableGraph Methods

The [PartitionIndexableGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------|---------------------------------------------|
|  SaveGraphson | (Defined by GraphHelpers.) |
|----------------------------------------------------------------------------------------------------------------|---------------------------------------------|

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph.CreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.CreateIndex (System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.CreateIndex (System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.CreateIndex
(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.CreateIndex
(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.CreateIndex
(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph.DropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.DropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropIndex(  
    string indexName  
)
```

VB

```
Public Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
) sealed
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.DropIndex(System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
) sealed
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndex(System.String,System.Type)"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndex(System.String,System.Type)"]

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionIndexableGraph.GetIndices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Function GetIndices As IEnumerable(Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndex^>^ GetIndices () sealed
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionIndexableGraph.GetIndices"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[PartitionIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class PartitionVertex : PartitionElement,
    IVertex, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class PartitionVertex
    Inherits PartitionElement
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class PartitionVertex : public PartitionElement,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type PartitionVertex =
    class
        inherit PartitionElement
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end


```

The **PartitionVertex** type exposes the following members.





Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|----------------------------------------------------------------|
|  | PartitionVertex | Initializes a new instance of the PartitionVertex class |







Properties






| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

Methods

| | Name | Description |
|------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are <code>equal()</code> . (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex Constructor

Initializes a new instance of the [PartitionVertex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public PartitionVertex(  
    IVertex vertex,  
    PartitionGraph innerTinkerGraph  
)
```

VB

```
Public Sub New (  
    vertex As IVertex,  
    innerTinkerGraph As PartitionGraph  
)
```

C++

```
public:  
PartitionVertex(  
    IVertex^ vertex,  
    PartitionGraph^ innerTinkerGraph  
)
```

F#

```
new :  
    vertex : IVertex *  
    innerTinkerGraph : PartitionGraph -> PartitionVertex
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"]

innerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph](#)

[Missing <param name="innerTinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionGraph)"]

VelocityDB Class Library

See Also


[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.PartitionVertex Properties

The [PartitionVertex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

See Also

[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.Vertex Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.Vertex"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex Vertex { get; }
```

VB

```
Public ReadOnly Property Vertex As IVertex  
    Get
```

C++

```
public:  
property IVertex^ Vertex {  
    IVertex^ get ();  
}
```

F#

```
member Vertex : IVertex with get
```

Property Value

Type: [IVertex](#)

See Also





[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)











PartitionVertex.PartitionVertex Methods

The [PartitionVertex](#) type exposes the following members.


Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  | VertexString | (Defined by StringFactory.) |

See Also

[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    string label,
    IVertex vertex
)
```

VB

```
Public Function AddEdge (
    id As Object,
    label As String,
    vertex As IVertex
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    String^ label,
    IVertex^ vertex
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
override AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also

[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetVertices (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IVertex)
```

C++

```
public:
virtual IEnumerable<IVertex^>^ GetVertices (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetVertices :
    direction : Direction *
    labels : string[] -> IEnumerable<IVertex>
override GetVertices :
    direction : Direction *
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[PartitionVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

PartitionVertex.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition.PartitionVertex.Query"]

Implements

[IVertex.Query\(\)](#)

See Also

[PartitionVertex Class](#)









[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Partition Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace

[Missing <summary> documentation for

"N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly"]

Classes

| Class | Description |
|-----------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  ReadOnlyEdge | |
|  ReadOnlyElement | |
|  ReadOnlyGraph | A ReadOnlyInnerTinkerGraph wraps a Graph and overrides the underlying Graph's mutating methods. In this way, a ReadOnlyInnerTinkerGraph can only be read from, not written to. |
|  ReadOnlyIndex | |
|  ReadOnlyIndexableGraph | |
|  ReadOnlyKeyIndexableGraph | A ReadOnlyKeyIndexableGraph wraps a KeyIndexableGraph and overrides the underlying graph's mutating methods. In this way, a ReadOnlyKeyIndexableGraph can only be read from, not written to. |
|  ReadOnlyTokens | |
|  ReadOnlyVertex | |

ReadOnlyEdge Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReadOnlyEdge : ReadOnlyElement,
    IEdge, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class ReadOnlyEdge
    Inherits ReadOnlyElement
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
    KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class ReadOnlyEdge : public ReadOnlyElement,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type ReadOnlyEdge =
    class
        inherit ReadOnlyElement
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end


```

The **ReadOnlyEdge** type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | ReadOnlyEdge | Initializes a new instance of the ReadOnlyEdge class |








Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | EdgeString | (Defined by StringFactory.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyEdge Constructor

Initializes a new instance of the [ReadOnlyEdge](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyEdge (
    ReadOnlyGraph innerTinkerGraph,
    IEdge baseEdge
)
```

VB

```
Public Sub New (
    innerTinkerGraph As ReadOnlyGraph,
    baseEdge As IEdge
)
```

C++

```
public:
    ReadOnlyEdge (
        ReadOnlyGraph^ innerTinkerGraph,
        IEdge^ baseEdge
    )
```

F#

```
new :
    innerTinkerGraph : ReadOnlyGraph *
    baseEdge : IEdge -> ReadOnlyEdge
```

Parameters

innerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph](#)

[Missing <param name="innerTinkerGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IEdge)"]

baseEdge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="baseEdge"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IEdge)"]

VelocityDB Class Library

See Also


[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyEdge.ReadOnlyEdge Properties

The [ReadOnlyEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Label | |

See Also

[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyEdge.Label Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Label { get; }
```

VB

```
Public ReadOnly Property Label As String  
    Get
```

C++

```
public:  
virtual property String^ Label {  
    String^ get () sealed;  
}
```

F#

```
abstract Label : string with get  
override Label : string with get
```

Property Value

Type: [String](#)

Implements

[IEdge.Label](#)

See Also

[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)











ReadOnlyEdge.ReadOnlyEdge Methods

The [ReadOnlyEdge](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | GetBaseEdge | |
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

VelocityDB Class Library

See Also

[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyEdge.GetBaseEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.GetBaseEdge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetBaseEdge ()
```

VB

```
Public Function GetBaseEdge As IEdge
```

C++

```
public:  
IEdge^ GetBaseEdge ()
```

F#

```
member GetBaseEdge : unit -> IEdge
```

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.GetBaseEdge"]

See Also

[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyEdge.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Public Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Direction direction  
) sealed
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex  
override GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IEdge.GetVertex\(Direction\)](#)

See Also

[ReadOnlyEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyEdge](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class ReadOnlyElement : DictionaryElement
```

VB

```
Public MustInherit Class ReadOnlyElement
    Inherits DictionaryElement
```

C++

```
public ref class ReadOnlyElement abstract : public DictionaryElement
```

F#


```
[<AbstractClassAttribute>]
type ReadOnlyElement =
    class
        inherit DictionaryElement
    end
```

The **ReadOnlyElement** type exposes the following members.







Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id .) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |

VelocityDB Class Library

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  Remove | (Overrides DictionaryElement.Remove().) |
|  RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  ToString | (Overrides Object.ToString().) |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.ReadOnlyElement Properties

The [ReadOnlyElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------|
|  | Id | (Overrides DictionaryElement.Id.) |

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.Id Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also









[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.ReadOnlyElement Methods

The [ReadOnlyElement](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys() .) |
|  | Remove | (Overrides DictionaryElement.Remove() .) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String) .) |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.Equals Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.Equals(System.Object)"]

See Also

[ReadOnlyElement Class](#)

VelocityDB Class Library

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.GetHashCode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetHashCode"]

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.GetProperty\(String\)](#)

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetPropertyKeys"
]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.GetPropertyKeys"
]

Implements

[IElement.GetPropertyKeys\(\)](#)

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.RemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.RemoveProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.RemoveProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.SetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.SetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.SetProperty(System.String,System.Object)"]

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyElement.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement.ToString"]

See Also

[ReadOnlyElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph Class

A ReadOnlyInnerTinkerGraph wraps a Graph and overrides the underlying Graph's mutating methods. In this way, a ReadOnlyInnerTinkerGraph can only be read from, not written to.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph
[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

```
C#
public class ReadOnlyGraph : IGraph,
    IWrapperGraph
```


```
VB
Public Class ReadOnlyGraph
    Implements IGraph, IWrapperGraph
```

```
C++
public ref class ReadOnlyGraph : IGraph,
    IWrapperGraph
```

```
F#
type ReadOnlyGraph =
    class
        interface IGraph
        interface IWrapperGraph
    end
```

The **ReadOnlyGraph** type exposes the following members.















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | ReadOnlyGraph | Initializes a new instance of the ReadOnlyGraph class |










Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |





Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------|
| | | mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph Constructor

Initializes a new instance of the [ReadOnlyGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyGraph(  
    IGraph baseGraph  
)
```

VB

```
Public Sub New (  
    baseGraph As IGraph  
)
```

C++

```
public:  
ReadOnlyGraph(  
    IGraph^ baseGraph  
)
```

F#

```
new :  
    baseGraph : IGraph -> ReadOnlyGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="baseGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also


[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.ReadOnlyGraph Properties

The [ReadOnlyGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also















[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)










ReadOnlyGraph.ReadOnlyGraph Methods





The [ReadOnlyGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.AddVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.AddVertex\(Object\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdge(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetEdge\(Object\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable (Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[ReadOnlyGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[ReadOnlyGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetVertex\(Object\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[ReadOnlyGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetVertices (  
    key As String,  
    value As Object  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>  
override GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[ReadOnlyGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (
    IEdge edge
)
```

VB

```
Public Sub RemoveEdge (
    edge As IEdge
)
```

C++

```
public:
virtual void RemoveEdge (
    IEdge^ edge
) sealed
```

F#

```
abstract RemoveEdge :
    edge : IEdge -> unit
override RemoveEdge :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph.ToString"]

See Also

[ReadOnlyGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReadOnlyIndex : IIndex
```

VB

```
Public Class ReadOnlyIndex
    Implements IIndex
```

C++


```
public ref class ReadOnlyIndex : IIndex
```

F#



```
type ReadOnlyIndex =
    class
        interface IIndex
    end
```

The **ReadOnlyIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | ReadOnlyIndex | Initializes a new instance of the ReadOnlyIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex Constructor

Initializes a new instance of the [ReadOnlyIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyIndex(  
    ReadOnlyGraph graph,  
    IIndex rawIndex  
)
```

VB

```
Public Sub New (  
    graph As ReadOnlyGraph,  
    rawIndex As IIndex  
)
```

C++

```
public:  
ReadOnlyIndex(  
    ReadOnlyGraph^ graph,  
    IIndex^ rawIndex  
)
```

F#

```
new :  
    graph : ReadOnlyGraph *  
    rawIndex : IIndex -> ReadOnlyIndex
```

Parameters

graph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph](#)

[Missing <param name="graph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IIndex)"]

rawIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="rawIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IIndex)"]

VelocityDB Class Library

See Also



[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.ReadOnlyIndex Properties

The [ReadOnlyIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Name Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Type Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.ReadOnlyIndex Methods

The [ReadOnlyIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Count Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Count(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Count(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Count(System.String, System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Count(System.String, System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Put(System.String, System.Object, VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Put(System.String, System.Object, VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Query(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Query(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Query(System.String, System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Query(System.String, System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndex.ToString"]

See Also

[ReadOnlyIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReadOnlyIndexableGraph : ReadOnlyGraph,
    IIndexableGraph, IGraph
```

VB

```
Public Class ReadOnlyIndexableGraph
    Inherits ReadOnlyGraph
    Implements IIndexableGraph, IGraph
```

C++


```
public ref class ReadOnlyIndexableGraph : public ReadOnlyGraph,
    IIndexableGraph, IGraph
```

F#





```
type ReadOnlyIndexableGraph =
    class
        inherit ReadOnlyGraph
        interface IIndexableGraph
        interface IGraph
    end
```

The **ReadOnlyIndexableGraph** type exposes the following members.















Constructors

| Name | Description |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
|  ReadOnlyIndexableGraph | Initializes a new instance of the ReadOnlyIndexableGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

VelocityDB Class Library

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph Constructor

Initializes a new instance of the [ReadOnlyIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyIndexableGraph(  
    IIndexableGraph baseIndexableGraph  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph  
)
```

C++

```
public:  
ReadOnlyIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph -> ReadOnlyIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph)"]

See Also





[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)














ReadOnlyIndexableGraph.ReadOnlyIndexableGraph Methods

The [ReadOnlyIndexableGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|
|  | SaveGraphson | (Defined by GraphHelpers.) |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|

See Also

[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph.CreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph.DropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.DropIndex (System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropIndex(  
    string indexName  
)
```

VB

```
Public Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
) sealed
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.DropIndex (System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
) sealed
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndex(System.String,System.Type)"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndex(System.String,System.Type)"]

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyIndexableGraph.GetIndices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndices
"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Function GetIndices As IEnumerable (Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndex^>^ GetIndices () sealed
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph.GetIndices
"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[ReadOnlyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyKeyIndexableGraph Class

A ReadOnlyKeyIndexableGraph wraps a KeyIndexableGraph and overrides the underlying graph's mutating methods. In this way, a ReadOnlyKeyIndexableGraph can only be read from, not written to.

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyIndexableGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReadOnlyKeyIndexableGraph : ReadOnlyIndexableGraph,
    IKeyIndexableGraph, IGraph
```

VB

```
Public Class ReadOnlyKeyIndexableGraph
    Inherits ReadOnlyIndexableGraph
    Implements IKeyIndexableGraph, IGraph
```

C++


```
public ref class ReadOnlyKeyIndexableGraph : public ReadOnlyIndexableGraph,
    IKeyIndexableGraph, IGraph
```

F#




```
type ReadOnlyKeyIndexableGraph =
    class
        inherit ReadOnlyIndexableGraph
        interface IKeyIndexableGraph
        interface IGraph
    end
```

The **ReadOnlyKeyIndexableGraph** type exposes the following members.














Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------------|--------------------------------------------------------------------------|
|  | ReadOnlyKeyIndexableGraph | Initializes a new instance of the ReadOnlyKeyIndexableGraph class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | CreateKeyIndex | |
|  | DropKeyIndex | |
|  | GetIndexedKeys | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyKeyIndexableGraph Constructor

Initializes a new instance of the [ReadOnlyKeyIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyKeyIndexableGraph(  
    IKeyIndexableGraph baseGraph  
)
```

VB

```
Public Sub New (  
    baseGraph As IKeyIndexableGraph  
)
```

C++

```
public:  
ReadOnlyKeyIndexableGraph(  
    IKeyIndexableGraph^ baseGraph  
)
```

F#

```
new :  
    baseGraph : IKeyIndexableGraph -> ReadOnlyKeyIndexableGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IKeyIndexableGraph](#)

[Missing <param name="baseGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IKeyIndexableGraph)"]

See Also




[ReadOnlyKeyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)














ReadOnlyKeyIndexableGraph.ReadOnlyKeyIndexableGraph Methods

The [ReadOnlyKeyIndexableGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|-------------|
|  | CreateKeyIndex | |
|  | DropKeyIndex | |
|  | GetIndexedKeys | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

See Also

[ReadOnlyKeyIndexableGraph Class](#)

ReadOnlyKeyIndexableGraph.CreateKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void CreateKeyIndex(  
    string key,  
    Type elementClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Sub CreateKeyIndex (  
    key As String,  
    elementClass As Type,  
    ParamArray indexParameters As Parameter()  
)
```

C++

```
public:  
virtual void CreateKeyIndex(  
    String^ key,  
    Type^ elementClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit  
override CreateKeyIndex :  
    key : string *  
    elementClass : Type *  
    indexParameters : Parameter[] -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.CreateKeyIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IKeyIndexableGraph.CreateKeyIndex\(String, Type, Parameter\[\]\)](#)

See Also

[ReadOnlyKeyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyKeyIndexableGraph.DropKeyIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.DropKeyIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropKeyIndex(  
    string key,  
    Type elementClass  
)
```

VB

```
Public Sub DropKeyIndex (  
    key As String,  
    elementClass As Type  
)
```

C++

```
public:  
virtual void DropKeyIndex(  
    String^ key,  
    Type^ elementClass  
) sealed
```

F#

```
abstract DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit  
override DropKeyIndex :  
    key : string *  
    elementClass : Type -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.DropKeyIndex(System.String,System.Type)"]

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.DropKeyIndex(System.String,System.Type)"]

Implements

[IKeyIndexableGraph.DropKeyIndex\(String, Type\)](#)

See Also

[ReadOnlyKeyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyKeyIndexableGraph.GetIndexedKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.GetIndexedKeys(System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<string> GetIndexedKeys (  
    Type elementClass  
)
```

VB

```
Public Function GetIndexedKeys (  
    elementClass As Type  
) As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetIndexedKeys (  
    Type^ elementClass  
) sealed
```

F#

```
abstract GetIndexedKeys :  
    elementClass : Type -> IEnumerable<string>  
override GetIndexedKeys :  
    elementClass : Type -> IEnumerable<string>
```

Parameters

elementClass

Type: [System.Type](#)

[Missing <param name="elementClass"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.GetIndexedKeys(System.Type)"]

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyKeyIndexableGraph.GetIndexedKeys(System.Type)"]

VelocityDB Class Library

Implements

[IKeyIndexableGraph.GetIndexedKeys\(Type\)](#)

See Also

[ReadOnlyKeyIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyTokens Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyTokens"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyTokens

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public static class ReadOnlyTokens
```

VB

```
Public NotInheritable Class ReadOnlyTokens
```

C++

```
public ref class ReadOnlyTokens abstract sealed
```

F#

```
[<AbstractClassAttribute>]  
[<SealedAttribute>]  
type ReadOnlyTokens = class end
```

The **ReadOnlyTokens** type exposes the following members.

Fields

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-------------|
|  | MutateErrorMessage | |


See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyTokens.ReadOnlyTokens Fields

The [ReadOnlyTokens](#) type exposes the following members.

Fields

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------|-------------|
|  | MutateErrorMessage | |

See Also

[ReadOnlyTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyTokens.MutateErrorMessage Field

[Missing <summary> documentation for

"F:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyTokens.MutateErrorMessage"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public const string MutateErrorMessage = "It is not possible to mutate a ReadOnlyInnerTinkerGraph"
```

VB

```
Public Const MutateErrorMessage As String = "It is not possible to mutate a ReadOnlyInnerTinkerGraph"
```

C++

```
public:  
literal String^ MutateErrorMessage = "It is not possible to mutate a ReadOnlyInnerTinkerGraph"
```

F#

```
static val mutable MutateErrorMessage: string
```

Field Value

Type: [String](#)

See Also

[ReadOnlyTokens Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class ReadOnlyVertex : ReadOnlyElement,
    IVertex, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class ReadOnlyVertex
    Inherits ReadOnlyElement
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
    KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class ReadOnlyVertex : public ReadOnlyElement,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type ReadOnlyVertex =
    class
        inherit ReadOnlyElement
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end





```

The **ReadOnlyVertex** type exposes the following members.









Constructors




| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------------|---------------------------------------------------------------|
|  | ReadOnlyVertex | Initializes a new instance of the ReadOnlyVertex class |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are <code>equal()</code> . (Defined by ElementHelpers.) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be <code>equals()</code> . (Defined by ElementHelpers.) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |

| | |
|---------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  VertexString | (Defined by StringFactory.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex Constructor

Initializes a new instance of the [ReadOnlyVertex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public ReadOnlyVertex(  
    ReadOnlyGraph innerTinkerGraph,  
    IVertex baseVertex  
)
```

VB

```
Public Sub New (  
    innerTinkerGraph As ReadOnlyGraph,  
    baseVertex As IVertex  
)
```

C++

```
public:  
ReadOnlyVertex(  
    ReadOnlyGraph^ innerTinkerGraph,  
    IVertex^ baseVertex  
)
```

F#

```
new :  
    innerTinkerGraph : ReadOnlyGraph *  
    baseVertex : IVertex -> ReadOnlyVertex
```

Parameters

innerTinkerGraph

Type: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph](#)

[Missing <param name="innerTinkerGraph"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IVertex)"]

baseVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="baseVertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.#ctor(VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyGraph,VelocityGraph.Frontenac.Blueprints.IVertex)"]

VelocityDB Class Library

See Also





[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)











ReadOnlyVertex.ReadOnlyVertex Methods

The [ReadOnlyVertex](#) type exposes the following members.


Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.) |
|  | VertexString | (Defined by StringFactory.) |

See Also

[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    string label,
    IVertex vertex
)
```

VB

```
Public Function AddEdge (
    id As Object,
    label As String,
    vertex As IVertex
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    String^ label,
    IVertex^ vertex
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
override AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also

[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Public Function GetVertices (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    Direction direction,  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>  
override GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

ReadOnlyVertex.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly.ReadOnlyVertex.Query"]

Implements

[IVertex.Query\(\)](#)

See Also







[ReadOnlyVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.ReadOnly Namespace](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace

[Missing <summary> documentation for "N:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped"]

Classes

| | Class | Description |
|-----------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|  | WrappedEdge | |
|  | WrappedElement | |
|  | WrappedGraph | WrappedGraph serves as a template for writing a wrapper graph. The intention is that the code in this template is copied and adjusted accordingly. |
|  | WrappedIndex | |
|  | WrappedIndexableGraph | |
|  | WrappedVertex | |

WrappedEdge Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappedEdge : WrappedElement,
    IEdge, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class WrappedEdge
    Inherits WrappedElement
    Implements IEdge, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class WrappedEdge : public WrappedElement,
    IEdge, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```

F#

```
type WrappedEdge =
    class
        inherit WrappedElement
        interface IEdge
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```

interface ICollection
end



```

The **WrappedEdge** type exposes the following members.

Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|------------------------------------------------------------|
|  | WrappedEdge | Initializes a new instance of the WrappedEdge class |








Properties




| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Edge | |
|  | Label | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | EdgeString | (Defined by StringFactory.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers.) |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedEdge Constructor

Initializes a new instance of the [WrappedEdge](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedEdge (
    IEdge edge
)
```

VB

```
Public Sub New (
    edge As IEdge
)
```

C++

```
public:
    WrappedEdge (
        IEdge^ edge
    )
```

F#

```
new :
    edge : IEdge -> WrappedEdge
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.#ctor(VelocityGraph.Frontenac.Blueprints.IEdge)"]

See Also



[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedEdge.WrappedEdge Properties

The [WrappedEdge](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------|-------------|
|  | Edge | |
|  | Label | |

See Also

[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedEdge.Edge Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.Edge"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge Edge { get; }
```

VB

```
Public ReadOnly Property Edge As IEdge  
    Get
```

C++

```
public:  
property IEdge^ Edge {  
    IEdge^ get ();  
}
```

F#

```
member Edge : IEdge with get
```

Property Value

Type: [IEdge](#)

See Also

[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedEdge.Label Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.Label"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Label { get; }
```

VB

```
Public ReadOnly Property Label As String  
    Get
```

C++

```
public:  
virtual property String^ Label {  
    String^ get () sealed;  
}
```

F#

```
abstract Label : string with get  
override Label : string with get
```

Property Value

Type: [String](#)

Implements

[IEdge.Label](#)

See Also

[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)











WrappedEdge.WrappedEdge Methods

The [WrappedEdge](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------|-------------|
|  | GetVertex | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | EdgeString | (Defined by StringFactory .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers .) |
|  | RelabelEdge | An edge is relabeled by creating a new edge with the same properties, but new label. Note that an edge is deleted and an edge is added. (Defined by EdgeHelpers .) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers .) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers .) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |

VelocityDB Class Library

See Also

[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedEdge.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Direction direction  
)
```

VB

```
Public Function GetVertex (  
    direction As Direction  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Direction direction  
) sealed
```

F#

```
abstract GetVertex :  
    direction : Direction -> IVertex  
override GetVertex :  
    direction : Direction -> IVertex
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge.GetVertex(VelocityGraph.Frontenac.Blueprints.Direction)"]

VelocityDB Class Library

Implements

[IEdge.GetVertex\(Direction\)](#)

See Also

[WrappedEdge Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement Class

[Missing <summary> documentation for

"T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedEdge](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public abstract class WrappedElement : DictionaryElement
```

VB

```
Public MustInherit Class WrappedElement
    Inherits DictionaryElement
```

C++



```
public ref class WrappedElement abstract : public DictionaryElement
```

F#


```
[<AbstractClassAttribute>]
type WrappedElement =
    class
        inherit DictionaryElement
    end
```

The **WrappedElement** type exposes the following members.








Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|-------------------------|----------------------------------------------------|
|  | Element | |
|  | Id | (Overrides DictionaryElement.Id .) |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------|-----------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | GetHashCode | (Overrides Object.GetHashCode().) |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String).) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys().) |
|  | Remove | (Overrides DictionaryElement.Remove().) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String).) |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object).) |
|  | ToString | (Overrides Object.ToString().) |



See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.WrappedElement Properties

The [WrappedElement](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------|----------------------------------------------------|
|  | Element | |
|  | Id | (Overrides DictionaryElement.Id.) |

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.Element Property

[Missing <summary> documentation for "P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Element"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IElement Element { get; }
```

VB

```
Public ReadOnly Property Element As IElement  
    Get
```

C++

```
public:  
property IElement^ Element {  
    IElement^ get ();  
}
```

F#

```
member Element : IElement with get
```

Property Value

Type: [IElement](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.Id Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Id"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object Id { get; }
```

VB

```
Public Overrides ReadOnly Property Id As Object  
    Get
```

C++

```
public:  
virtual property Object^ Id {  
    Object^ get () override;  
}
```

F#

```
abstract Id : Object with get  
override Id : Object with get
```

Property Value

Type: [Object](#)

Implements

[IElement.Id](#)

See Also









[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.WrappedElement Methods

The [WrappedElement](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------------------------------|
|  | Equals | (Overrides Object.Equals(Object) .) |
|  | GetHashCode | (Overrides Object.GetHashCode() .) |
|  | GetProperty | (Overrides DictionaryElement.GetProperty(String) .) |
|  | GetPropertyKeys | (Overrides DictionaryElement.GetPropertyKeys() .) |
|  | Remove | (Overrides DictionaryElement.Remove() .) |
|  | RemoveProperty | (Overrides DictionaryElement.RemoveProperty(String) .) |
|  | SetProperty | (Overrides DictionaryElement.SetProperty(String, Object) .) |
|  | ToString | (Overrides Object.ToString() .) |

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.Equals Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Equals(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override bool Equals(  
    Object obj  
)
```

VB

```
Public Overrides Function Equals (  
    obj As Object  
) As Boolean
```

C++

```
public:  
virtual bool Equals(  
    Object^ obj  
) override
```

F#

```
abstract Equals :  
    obj : Object -> bool  
override Equals :  
    obj : Object -> bool
```

Parameters

obj

Type: [System.Object](#)

[Missing <param name="obj"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Equals(System.Object)"]

Return Value

Type: [Boolean](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Equals(System.Object)"]

See Also

[WrappedElement Class](#)

WrappedElement.GetHashCode Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetHashCode"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override int GetHashCode()
```

VB

```
Public Overrides Function GetHashCode As Integer
```

C++

```
public:  
virtual int GetHashCode() override
```

F#

```
abstract GetHashCode : unit -> int  
override GetHashCode : unit -> int
```

Return Value

Type: [Int32](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetHashCode"]

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.GetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object GetProperty(  
    string key  
)
```

VB

```
Public Overrides Function GetProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ GetProperty(  
    String^ key  
) override
```

F#

```
abstract GetProperty :  
    key : string -> Object  
override GetProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.GetProperty\(String\)](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.GetPropertyKeys Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetPropertyKeys"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override IEnumerable<string> GetPropertyKeys ()
```

VB

```
Public Overrides Function GetPropertyKeys As IEnumerable(Of String)
```

C++

```
public:  
virtual IEnumerable<String^>^ GetPropertyKeys () override
```

F#

```
abstract GetPropertyKeys : unit -> IEnumerable<string>  
override GetPropertyKeys : unit -> IEnumerable<string>
```

Return Value

Type: [IEnumerable\(String\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.GetPropertyKeys"]

Implements

[IElement.GetPropertyKeys\(\)](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.Remove Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.Remove"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void Remove ()
```

VB

```
Public Overrides Sub Remove
```

C++

```
public:  
virtual void Remove () override
```

F#

```
abstract Remove : unit -> unit  
override Remove : unit -> unit
```

Implements

[IElement.Remove\(\)](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.RemoveProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.RemoveProperty(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override Object RemoveProperty(  
    string key  
)
```

VB

```
Public Overrides Function RemoveProperty (  
    key As String  
) As Object
```

C++

```
public:  
virtual Object^ RemoveProperty(  
    String^ key  
) override
```

F#

```
abstract RemoveProperty :  
    key : string -> Object  
override RemoveProperty :  
    key : string -> Object
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.RemoveProperty(System.String)"]

Return Value

Type: [Object](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.RemoveProperty(System.String)"]

VelocityDB Class Library

Implements

[IElement.RemoveProperty\(String\)](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.SetProperty Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.SetProperty(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override void SetProperty(  
    string key,  
    Object value  
)
```

VB

```
Public Overrides Sub SetProperty (  
    key As String,  
    value As Object  
)
```

C++

```
public:  
virtual void SetProperty(  
    String^ key,  
    Object^ value  
) override
```

F#

```
abstract SetProperty :  
    key : string *  
    value : Object -> unit  
override SetProperty :  
    key : string *  
    value : Object -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.SetProperty(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.SetProperty(System.String,System.Object)"]

Implements

[IElement.SetProperty\(String, Object\)](#)

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedElement.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement.ToString"]

See Also

[WrappedElement Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph Class

WrappedGraph serves as a template for writing a wrapper graph. The intention is that the code in this template is copied and adjusted accordingly.

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph](#)

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappedGraph : IGraph,
    IWrapperGraph
```

VB

```
Public Class WrappedGraph
    Implements IGraph, IWrapperGraph
```

C++


```
public ref class WrappedGraph : IGraph,
    IWrapperGraph
```

F#


```
type WrappedGraph =
    class
        interface IGraph
        interface IWrapperGraph
    end
```

The **WrappedGraph** type exposes the following members.















Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | WrappedGraph | Initializes a new instance of the WrappedGraph class |










Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |





Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------|
| | | mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph Constructor

Initializes a new instance of the [WrappedGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedGraph(  
    IGraph baseGraph  
)
```

VB

```
Public Sub New (  
    baseGraph As IGraph  
)
```

C++

```
public:  
WrappedGraph(  
    IGraph^ baseGraph  
)
```

F#

```
new :  
    baseGraph : IGraph -> WrappedGraph
```

Parameters

baseGraph

Type: [VelocityGraph.Frontenac.Blueprints.IGraph](#)

[Missing <param name="baseGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IGraph)"]

See Also


[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.WrappedGraph Properties

The [WrappedGraph](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------|
|  | Features | |

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.Features Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.Features"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Features Features { get; }
```

VB

```
Public ReadOnly Property Features As Features  
    Get
```

C++

```
public:  
virtual property Features^ Features {  
    Features^ get () sealed;  
}
```

F#

```
abstract Features : Features with get  
override Features : Features with get
```

Property Value

Type: [Features](#)

Implements

[IGraph.Features](#)

See Also















[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)










WrappedGraph.WrappedGraph Methods





The [WrappedGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------|
|  | AddEdge | |
|  | AddVertex | |
|  | GetBaseGraph | |
|  | GetEdge | |
|  | GetEdges() | |
|  | GetEdges(String, Object) | |
|  | GetVertex | |
|  | GetVertices() | |
|  | GetVertices(String, Object) | |
|  | Query | |
|  | RemoveEdge | |
|  | RemoveVertex | |
|  | Shutdown | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers.) |
|  | SaveDotNet | (Defined by GraphHelpers.) |
|  | SaveGml | (Defined by GraphHelpers.) |
|  | SaveGraphml | (Defined by GraphHelpers.) |
|  | SaveGraphson | (Defined by GraphHelpers.) |

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    IVertex outVertex,
    IVertex inVertex,
    string label
)
```

VB

```
Public Function AddEdge (
    id As Object,
    outVertex As IVertex,
    inVertex As IVertex,
    label As String
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    IVertex^ outVertex,
    IVertex^ inVertex,
    String^ label
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
override AddEdge :
    id : Object *
    outVertex : IVertex *
    inVertex : IVertex *
    label : string -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

outVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="outVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

inVertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="inVertex"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddEdge(System.Object,VelocityGraph.Frontenac.Blueprints.IVertex,VelocityGraph.Frontenac.Blueprints.IVertex,System.String)"]

Implements

[IGraph.AddEdge\(Object, IVertex, IVertex, String\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.AddVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex AddVertex(  
    Object id  
)
```

VB

```
Public Function AddVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ AddVertex(  
    Object^ id  
) sealed
```

F#

```
abstract AddVertex :  
    id : Object -> IVertex  
override AddVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.AddVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.AddVertex\(Object\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetBaseGraph Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetBaseGraph"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IGraph GetBaseGraph()
```

VB

```
Public Function GetBaseGraph As IGraph
```

C++

```
public:  
virtual IGraph^ GetBaseGraph() sealed
```

F#

```
abstract GetBaseGraph : unit -> IGraph  
override GetBaseGraph : unit -> IGraph
```

Return Value

Type: [IGraph](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetBaseGraph"]

Implements

[IWrapperGraph.GetBaseGraph\(\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdge(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge GetEdge (
    Object id
)
```

VB

```
Public Function GetEdge (
    id As Object
) As IEdge
```

C++

```
public:
virtual IEdge^ GetEdge (
    Object^ id
) sealed
```

F#

```
abstract GetEdge :
    id : Object -> IEdge
override GetEdge :
    id : Object -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdge(System.Object)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdge(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetEdge\(Object\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetEdges Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------------------------|-------------|
|  | GetEdges() | |
|  | GetEdges(String, Object) | |

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges ()
```

VB

```
Public Function GetEdges As IEnumerable (Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges () sealed
```

F#

```
abstract GetEdges : unit -> IEnumerable<IEdge>  
override GetEdges : unit -> IEnumerable<IEdge>
```

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges"]

Implements

[IGraph.GetEdges\(\)](#)

See Also

[WrappedGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetEdges Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetEdges (  
    key As String,  
    value As Object  
) As IEnumerable(Of IEdge)
```

C++

```
public:  
virtual IEnumerable<IEdge^>^ GetEdges (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>  
override GetEdges :  
    key : string *  
    value : Object -> IEnumerable<IEdge>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetEdges(System.String,System.Object)"]

Implements

[IGraph.GetEdges\(String, Object\)](#)

See Also

[WrappedGraph Class](#)

[GetEdges Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertex(System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex GetVertex(  
    Object id  
)
```

VB

```
Public Function GetVertex (  
    id As Object  
) As IVertex
```

C++

```
public:  
virtual IVertex^ GetVertex(  
    Object^ id  
) sealed
```

F#

```
abstract GetVertex :  
    id : Object -> IVertex  
override GetVertex :  
    id : Object -> IVertex
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertex(System.Object)"]

Return Value

Type: [IVertex](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertex(System.Object)"]

VelocityDB Class Library

Implements

[IGraph.GetVertex\(Object\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetVertices Method

Overload List

| | Name | Description |
|-----------------------------------------------------------------------------------|---------------------------------------------|-------------|
|  | GetVertices() | |
|  | GetVertices(String, Object) | |

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices ()
```

VB

```
Public Function GetVertices As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices () sealed
```

F#

```
abstract GetVertices : unit -> IEnumerable<IVertex>  
override GetVertices : unit -> IEnumerable<IVertex>
```

Return Value

Type: [IEnumerable<IVertex>](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices"]

Implements

[IGraph.GetVertices\(\)](#)

See Also

[WrappedGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.GetVertices Method (String, Object)

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    string key,  
    Object value  
)
```

VB

```
Public Function GetVertices (  
    key As String,  
    value As Object  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>  
override GetVertices :  
    key : string *  
    value : Object -> IEnumerable<IVertex>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.GetVertices(System.String,System.Object)"]

Implements

[IGraph.GetVertices\(String, Object\)](#)

See Also

[WrappedGraph Class](#)

[GetVertices Overload](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IQuery Query()
```

VB

```
Public Function Query As IQuery
```

C++

```
public:  
virtual IQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IQuery  
override Query : unit -> IQuery
```

Return Value

Type: [IQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.Query"]

Implements

[IGraph.Query\(\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.RemoveEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveEdge (
    IEdge edge
)
```

VB

```
Public Sub RemoveEdge (
    edge As IEdge
)
```

C++

```
public:
virtual void RemoveEdge (
    IEdge^ edge
) sealed
```

F#

```
abstract RemoveEdge :
    edge : IEdge -> unit
override RemoveEdge :
    edge : IEdge -> unit
```

Parameters

edge

Type: [VelocityGraph.Frontenac.Blueprints.IEdge](#)

[Missing <param name="edge"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.RemoveEdge(VelocityGraph.Frontenac.Blueprints.IEdge)"]

Implements

[IGraph.RemoveEdge\(IEdge\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.RemoveVertex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void RemoveVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub RemoveVertex (  
    vertex As IVertex  
)
```

C++

```
public:  
virtual void RemoveVertex(  
    IVertex^ vertex  
) sealed
```

F#

```
abstract RemoveVertex :  
    vertex : IVertex -> unit  
override RemoveVertex :  
    vertex : IVertex -> unit
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.RemoveVertex(VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IGraph.RemoveVertex\(IVertex\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.Shutdown Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.Shutdown"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Shutdown ()
```

VB

```
Public Sub Shutdown
```

C++

```
public:  
virtual void Shutdown () sealed
```

F#

```
abstract Shutdown : unit -> unit  
override Shutdown : unit -> unit
```

Implements

[IGraph.Shutdown\(\)](#)

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedGraph.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph.ToString"]

See Also

[WrappedGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex"]

Inheritance Hierarchy

[System.Object](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappedIndex : IIndex
```

VB

```
Public Class WrappedIndex  
    Implements IIndex
```

C++


```
public ref class WrappedIndex : IIndex
```

F#



```
type WrappedIndex =  
    class  
        interface IIndex  
    end
```

The **WrappedIndex** type exposes the following members.







Constructors

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------|
|  | WrappedIndex | Initializes a new instance of the WrappedIndex class |

Properties

| | Name | Description |
|-------------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex Constructor

Initializes a new instance of the [WrappedIndex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedIndex (  
    IIndex rawIndex  
)
```

VB

```
Public Sub New (  
    rawIndex As IIndex  
)
```

C++

```
public:  
WrappedIndex (  
    IIndex^ rawIndex  
)
```

F#

```
new :  
    rawIndex : IIndex -> WrappedIndex
```

Parameters

rawIndex

Type: [VelocityGraph.Frontenac.Blueprints.IIndex](#)

[Missing <param name="rawIndex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.#ctor(VelocityGraph.Frontenac.Blueprints.IIndex)"]

See Also



[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.WrappedIndex Properties

The [WrappedIndex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|----------------------|-------------|
|  | Name | |
|  | Type | |

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Name Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Name"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public string Name { get; }
```

VB

```
Public ReadOnly Property Name As String  
    Get
```

C++

```
public:  
virtual property String^ Name {  
    String^ get () sealed;  
}
```

F#

```
abstract Name : string with get  
override Name : string with get
```

Property Value

Type: [String](#)

Implements

[IIndex.Name](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Type Property

[Missing <summary> documentation for
"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Type"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public Type Type { get; }
```

VB

```
Public ReadOnly Property Type As Type  
    Get
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () sealed;  
}
```

F#

```
abstract Type : Type with get  
override Type : Type with get
```

Property Value

Type: [Type](#)

Implements

[IIndex.Type](#)

See Also







[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.WrappedIndex Methods

The [WrappedIndex](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|--------------------------|-------------------------------------------------|
|  | Count | |
|  | Get | |
|  | Put | |
|  | Query | |
|  | Remove | |
|  | ToString | (Overrides Object.ToString() .) |

Extension Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|----------------------------------------------|
|  | IndexString | (Defined by StringFactory .) |

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Count Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Count(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public long Count(  
    string key,  
    Object value  
)
```

VB

```
Public Function Count (  
    key As String,  
    value As Object  
) As Long
```

C++

```
public:  
virtual long long Count(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Count :  
    key : string *  
    value : Object -> int64  
override Count :  
    key : string *  
    value : Object -> int64
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Count(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Count(System.String, System.Object)"]

Return Value

Type: [Int64](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Count(System.String, System.Object)"]

Implements

[IIndex.Count\(String, Object\)](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Get Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Get(System.String,System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Get (
    string key,
    Object value
)
```

VB

```
Public Function Get (
    key As String,
    value As Object
) As IEnumerable(Of IElement)
```

C++

```
public:
virtual IEnumerable<IElement^>^ Get (
    String^ key,
    Object^ value
) sealed
```

F#

```
abstract Get :
    key : string *
    value : Object -> IEnumerable<IElement>
override Get :
    key : string *
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Get(System.String,System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Get(System.String,System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Get(System.String,System.Object)"]

Implements

[IIndex.Get\(String, Object\)](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Put Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Put (  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Put (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Put(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Put :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Put(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Put\(String, Object, IElement\)](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Query Method

[Missing <summary> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Query(System.String, System.Object)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IElement> Query(  
    string key,  
    Object value  
)
```

VB

```
Public Function Query (  
    key As String,  
    value As Object  
) As IEnumerable(Of IElement)
```

C++

```
public:  
virtual IEnumerable<IElement^>^ Query(  
    String^ key,  
    Object^ value  
) sealed
```

F#

```
abstract Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>  
override Query :  
    key : string *  
    value : Object -> IEnumerable<IElement>
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Query(System.String, System.Object)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Query(System.String, System.Object)"]

Return Value

Type: [IEnumerable\(IElement\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Query(System.String, System.Object)"]

Implements

[IIndex.Query\(String, Object\)](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.Remove Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void Remove(  
    string key,  
    Object value,  
    IElement element  
)
```

VB

```
Public Sub Remove (  
    key As String,  
    value As Object,  
    element As IElement  
)
```

C++

```
public:  
virtual void Remove(  
    String^ key,  
    Object^ value,  
    IElement^ element  
) sealed
```

F#

```
abstract Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit  
override Remove :  
    key : string *  
    value : Object *  
    element : IElement -> unit
```

Parameters

key

Type: [System.String](#)

[Missing <param name="key"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

value

Type: [System.Object](#)

[Missing <param name="value"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

element

Type: [VelocityGraph.Frontenac.Blueprints.IElement](#)

[Missing <param name="element"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.Remove(System.String,System.Object,VelocityGraph.Frontenac.Blueprints.IElement)"]

Implements

[IIndex.Remove\(String, Object, IElement\)](#)

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndex.ToString Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.ToString"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public override string ToString()
```

VB

```
Public Overrides Function ToString As String
```

C++

```
public:  
virtual String^ ToString() override
```

F#

```
abstract ToString : unit -> string  
override ToString : unit -> string
```

Return Value

Type: [String](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndex.ToString"]

See Also

[WrappedIndex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph Class

[Missing <summary> documentation for "[T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph](#)"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedGraph](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappedIndexableGraph : WrappedGraph,
    IIndexableGraph, IGraph
```

VB

```
Public Class WrappedIndexableGraph
    Inherits WrappedGraph
    Implements IIndexableGraph, IGraph
```

C++


```
public ref class WrappedIndexableGraph : public WrappedGraph,
    IIndexableGraph, IGraph
```

F#

```
type WrappedIndexableGraph =
    class
        inherit WrappedGraph
        interface IIndexableGraph
        interface IGraph
    end
```

The **WrappedIndexableGraph** type exposes the following members.

Constructors















| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------|
|  | WrappedIndexableGraph | Initializes a new instance of the WrappedIndexableGraph class |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |

| | | |
|-----------------------------------------------------------------------------------|----------------------------|--|
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |
|  | SaveGraphson | (Defined by GraphHelpers .) |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph Constructor

Initializes a new instance of the [WrappedIndexableGraph](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedIndexableGraph(  
    IIndexableGraph baseIndexableGraph  
)
```

VB

```
Public Sub New (  
    baseIndexableGraph As IIndexableGraph  
)
```

C++

```
public:  
WrappedIndexableGraph(  
    IIndexableGraph^ baseIndexableGraph  
)
```

F#

```
new :  
    baseIndexableGraph : IIndexableGraph -> WrappedIndexableGraph
```

Parameters

baseIndexableGraph

Type: [VelocityGraph.Frontenac.Blueprints.IIndexableGraph](#)

[Missing <param name="baseIndexableGraph"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.#ctor(VelocityGraph.Frontenac.Blueprints.IIndexableGraph)"]

See Also





[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)














WrappedIndexableGraph.WrappedIndexableGraph Methods

The [WrappedIndexableGraph](#) type exposes the following members.

Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | CreateIndex | |
|  | DropIndex | |
|  | GetIndex | |
|  | GetIndices | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AddEdge | Add an edge to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | AddUniqueVertex | Add a vertex to a graph only if no other vertex in the provided Index is indexed by the property key/value pair. If a vertex already exists with that key/value pair, return the pre-existing vertex. (Defined by IndexableGraphHelpers .) |
|  | AddVertex | Add a vertex to the graph with specified id and provided properties. (Defined by GraphHelpers .) |
|  | CopyGraph | Copy the vertex/edges of one graph over to another graph. The id of the elements in the from graph are attempted to be used in the to graph. This method only works for graphs where the user can control the element ids. (Defined by GraphHelpers .) |
|  | CreateTinkerGraph | (Defined by GraphHelpers .) |
|  | GraphString | (Defined by StringFactory .) |
|  | LoadGml | (Defined by GraphHelpers .) |
|  | LoadGraphml | (Defined by GraphHelpers .) |
|  | LoadGraphson | (Defined by GraphHelpers .) |
|  | ReIndexElements(T) | For those graphs that do not support automatic reindexing of elements when a key is provided for indexing, this method can be used to simulate that behavior. The elements in the graph are iterated and their properties (for the provided keys) are removed and then added. Be sure that the key indices have been created prior to calling this method so that they can pick up the property mutations calls. Finally, if the graph is a TransactionalGraph, then a 1000 mutation buffer is used for each commit. (Defined by KeyIndexableGraphHelpers .) |
|  | SaveDotNet | (Defined by GraphHelpers .) |
|  | SaveGml | (Defined by GraphHelpers .) |
|  | SaveGraphml | (Defined by GraphHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|
|  | SaveGraphson | (Defined by GraphHelpers.) |
|-----------------------------------------------------------------------------------|------------------------------|---------------------------------------------|

See Also

[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph.CreateIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex CreateIndex(  
    string indexName,  
    Type indexClass,  
    params Parameter[] indexParameters  
)
```

VB

```
Public Function CreateIndex (  
    indexName As String,  
    indexClass As Type,  
    ParamArray indexParameters As Parameter()  
) As IIndex
```

C++

```
public:  
virtual IIndex^ CreateIndex(  
    String^ indexName,  
    Type^ indexClass,  
    ... array<Parameter^>^ indexParameters  
) sealed
```

F#

```
abstract CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex  
override CreateIndex :  
    indexName : string *  
    indexClass : Type *  
    indexParameters : Parameter[] -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

indexParameters

Type: [VelocityGraph.Frontenac.Blueprints.Parameter\[\]](#)

[Missing <param name="indexParameters"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.CreateIndex(System.String,System.Type,VelocityGraph.Frontenac.Blueprints.Parameter[])"]

Implements

[IIndexableGraph.CreateIndex\(String, Type,Parameter\[\]\)](#)

See Also

[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph.DropIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.DropIndex(System.String)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public void DropIndex(  
    string indexName  
)
```

VB

```
Public Sub DropIndex (  
    indexName As String  
)
```

C++

```
public:  
virtual void DropIndex(  
    String^ indexName  
) sealed
```

F#

```
abstract DropIndex :  
    indexName : string -> unit  
override DropIndex :  
    indexName : string -> unit
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.DropIndex(System.String)"]

Implements

[IIndexableGraph.DropIndex\(String\)](#)

See Also

[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph.GetIndex Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndex(System.String,System.Type)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IIndex GetIndex(  
    string indexName,  
    Type indexClass  
)
```

VB

```
Public Function GetIndex (  
    indexName As String,  
    indexClass As Type  
) As IIndex
```

C++

```
public:  
virtual IIndex^ GetIndex(  
    String^ indexName,  
    Type^ indexClass  
) sealed
```

F#

```
abstract GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex  
override GetIndex :  
    indexName : string *  
    indexClass : Type -> IIndex
```

Parameters

indexName

Type: [System.String](#)

[Missing <param name="indexName"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndex(System.String,System.Type)"]

indexClass

Type: [System.Type](#)

[Missing <param name="indexClass"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndex(System.String,System.Type)"]

Return Value

Type: [IIndex](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndex(System.String,System.Type)"]

Implements

[IIndexableGraph.GetIndex\(String, Type\)](#)

See Also

[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedIndexableGraph.GetIndices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndices"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IIndex> GetIndices ()
```

VB

```
Public Function GetIndices As IEnumerable (Of IIndex)
```

C++

```
public:  
virtual IEnumerable<IIndex^>^ GetIndices () sealed
```

F#

```
abstract GetIndices : unit -> IEnumerable<IIndex>  
override GetIndices : unit -> IEnumerable<IIndex>
```

Return Value

Type: [IEnumerable\(IIndex\)](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedIndexableGraph.GetIndices"]

Implements

[IIndexableGraph.GetIndices\(\)](#)

See Also

[WrappedIndexableGraph Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex Class

[Missing <summary> documentation for "T:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex"]

Inheritance Hierarchy

[System.Object](#)

[VelocityGraph.Frontenac.Blueprints.DictionaryElement](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedElement](#)

VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public class WrappedVertex : WrappedElement,
    IVertex, IElement, IDictionary<string, Object>,
    ICollection<KeyValuePair<string, Object>>,
    IEnumerable<KeyValuePair<string, Object>>,
    IEnumerable, IDictionary, ICollection
```

VB

```
Public Class WrappedVertex
    Inherits WrappedElement
    Implements IVertex, IElement, IDictionary(Of String, Object),
    ICollection(Of KeyValuePair(Of String, Object)), IEnumerable(Of
KeyValuePair(Of String, Object)),
    IEnumerable, IDictionary, ICollection
```

C++

```
public ref class WrappedVertex : public WrappedElement,
    IVertex, IElement, IDictionary<String^, Object^>,
    ICollection<KeyValuePair<String^, Object^>>,
    IEnumerable<KeyValuePair<String^, Object^>>,
    IEnumerable, IDictionary, ICollection
```


F#

```
type WrappedVertex =
    class
        inherit WrappedElement
        interface IVertex
        interface IElement
        interface IDictionary<string, Object>
        interface ICollection<KeyValuePair<string, Object>>
        interface IEnumerable<KeyValuePair<string, Object>>
        interface IEnumerable
        interface IDictionary
```


```
interface ICollection
end
```

The **WrappedVertex** type exposes the following members.





Constructors

| | Name | Description |
|-----------------------------------------------------------------------------------|-------------------------------|--------------------------------------------------------------|
|  | WrappedVertex | Initializes a new instance of the WrappedVertex class |







Properties






| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers .) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers .) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers .) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers .) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers .) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers .) |

| | | |
|-----------------------------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | HaveEqualProperties | <p>Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.)</p> |
|  | SetProperties(IDictionary(String, Object)) | <p>Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.)</p> |
|  | SetProperties(Object[]) | <p>Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.)</p> |
|  | ValidateProperty | <p>Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws <code>ArgumentException</code> whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers.)</p> |
|  | VertexString | <p>(Defined by StringFactory.)</p> |

See Also

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex Constructor

Initializes a new instance of the [WrappedVertex](#) class

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public WrappedVertex(  
    IVertex vertex  
)
```

VB

```
Public Sub New (  
    vertex As IVertex  
)
```

C++

```
public:  
WrappedVertex(  
    IVertex^ vertex  
)
```

F#

```
new :  
    vertex : IVertex -> WrappedVertex
```

Parameters

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.#ctor(VelocityGraph.Frontenac.Blueprints.IVertex)"]

See Also


[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.WrappedVertex Properties

The [WrappedVertex](#) type exposes the following members.

Properties

| | Name | Description |
|-----------------------------------------------------------------------------------|------------------------|-------------|
|  | Vertex | |

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.Vertex Property

[Missing <summary> documentation for

"P:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.Vertex"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertex Vertex { get; }
```

VB

```
Public ReadOnly Property Vertex As IVertex  
    Get
```

C++

```
public:  
property IVertex^ Vertex {  
    IVertex^ get ();  
}
```

F#

```
member Vertex : IVertex with get
```

Property Value

Type: [IVertex](#)

See Also





[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)











WrappedVertex.WrappedVertex Methods

The [WrappedVertex](#) type exposes the following members.


Methods

| | Name | Description |
|-----------------------------------------------------------------------------------|-----------------------------|-------------|
|  | AddEdge | |
|  | GetEdges | |
|  | GetVertices | |
|  | Query | |

Extension Methods

| | Name | Description |
|-------------------------------------------------------------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | AreEqual | A standard method for determining if two elements are equal. This method should be used by any <code>Element.equals()</code> implementation to ensure consistent behavior. (Defined by ElementHelpers.) |
|  | CopyProperties | Copy the properties (key and value) from one element to another. The properties are preserved on the from element. <code>ElementPropertiesRule</code> that share the same key on the to element are overwritten. (Defined by ElementHelpers.) |
|  | GetProperties | Get a clone of the properties of the provided element. In other words, a <code>HashMap</code> is created and filled with the key/values of the element's properties. (Defined by ElementHelpers.) |
|  | HaveEqualEdges | Test whether the two vertices have equal edge sets (Defined by VertexHelpers.) |
|  | HaveEqualIds | Simply tests if the element ids are equal(). (Defined by ElementHelpers.) |
|  | HaveEqualNeighborhood | Test whether the two vertices have equal properties and edge sets. (Defined by VertexHelpers.) |
|  | HaveEqualProperties | Determines whether two elements have the same properties. To be true, both must have the same property keys and respective values must be equals(). (Defined by ElementHelpers.) |
|  | SetProperties(IDictionary(String, Object)) | Overloaded. Set the properties of the provided element using the provided dictionary. (Defined by ElementHelpers.) |
|  | SetProperties(Object[]) | Overloaded. Set the properties of the provided element using the provided key value pairs. The var args of Objects must be divisible by 2. All odd elements in the array must be a string key. (Defined by ElementHelpers.) |
|  | ValidateProperty | Determines whether the property key/value for the specified element can be legally set. This is typically used as a pre-condition check prior to setting a property. Throws |

VelocityDB Class Library

| | | |
|-----------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| | | ArgumentException whether the triple is legal and if not, a clear reason message is provided (Defined by ElementHelpers .) |
|  | VertexString | (Defined by StringFactory .) |

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.AddEdge Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEdge AddEdge (
    Object id,
    string label,
    IVertex vertex
)
```

VB

```
Public Function AddEdge (
    id As Object,
    label As String,
    vertex As IVertex
) As IEdge
```

C++

```
public:
virtual IEdge^ AddEdge (
    Object^ id,
    String^ label,
    IVertex^ vertex
) sealed
```

F#

```
abstract AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
override AddEdge :
    id : Object *
    label : string *
    vertex : IVertex -> IEdge
```

Parameters

id

Type: [System.Object](#)

[Missing <param name="id"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

label

Type: [System.String](#)

[Missing <param name="label"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

vertex

Type: [VelocityGraph.Frontenac.Blueprints.IVertex](#)

[Missing <param name="vertex"/> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Return Value

Type: [IEdge](#)

[Missing <returns> documentation for
"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.AddEdge(System.Object,System.String,VelocityGraph.Frontenac.Blueprints.IVertex)"]

Implements

[IVertex.AddEdge\(Object, String, IVertex\)](#)

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.GetEdges Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IEdge> GetEdges (
    Direction direction,
    params string[] labels
)
```

VB

```
Public Function GetEdges (
    direction As Direction,
    ParamArray labels As String()
) As IEnumerable(Of IEdge)
```

C++

```
public:
virtual IEnumerable<IEdge^>^ GetEdges (
    Direction direction,
    ... array<String^>^ labels
) sealed
```

F#

```
abstract GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
override GetEdges :
    direction : Direction *
    labels : string[] -> IEnumerable<IEdge>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IEdge\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetEdges(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetEdges\(Direction,String\[\]\)](#)

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.GetVertices Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IEnumerable<IVertex> GetVertices (  
    Direction direction,  
    params string[] labels  
)
```

VB

```
Public Function GetVertices (  
    direction As Direction,  
    ParamArray labels As String()  
) As IEnumerable(Of IVertex)
```

C++

```
public:  
virtual IEnumerable<IVertex^>^ GetVertices (  
    Direction direction,  
    ... array<String^>^ labels  
) sealed
```

F#

```
abstract GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>  
override GetVertices :  
    direction : Direction *  
    labels : string[] -> IEnumerable<IVertex>
```

Parameters

direction

Type: [VelocityGraph.Frontenac.Blueprints.Direction](#)

[Missing <param name="direction"/> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

labels

Type: [System.String\[\]](#)

[Missing <param name="labels"/> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Return Value

Type: [IEnumerable\(IVertex\)](#)

[Missing <returns> documentation for "M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.GetVertices(VelocityGraph.Frontenac.Blueprints.Direction,System.String[])"]

Implements

[IVertex.GetVertices\(Direction,String\[\]\)](#)

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)

WrappedVertex.Query Method

[Missing <summary> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.Query"]

Namespace: [VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped](#)

Assembly: VelocityGraph (in VelocityGraph.dll) Version: 11.1.0.0 (11.1)

Syntax

C#

```
public IVertexQuery Query()
```

VB

```
Public Function Query As IVertexQuery
```

C++

```
public:  
virtual IVertexQuery^ Query() sealed
```

F#

```
abstract Query : unit -> IVertexQuery  
override Query : unit -> IVertexQuery
```

Return Value

Type: [IVertexQuery](#)

[Missing <returns> documentation for

"M:VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped.WrappedVertex.Query"]

Implements

[IVertex.Query\(\)](#)

See Also

[WrappedVertex Class](#)

[VelocityGraph.Frontenac.Blueprints.Util.Wrappers.Wrapped Namespace](#)