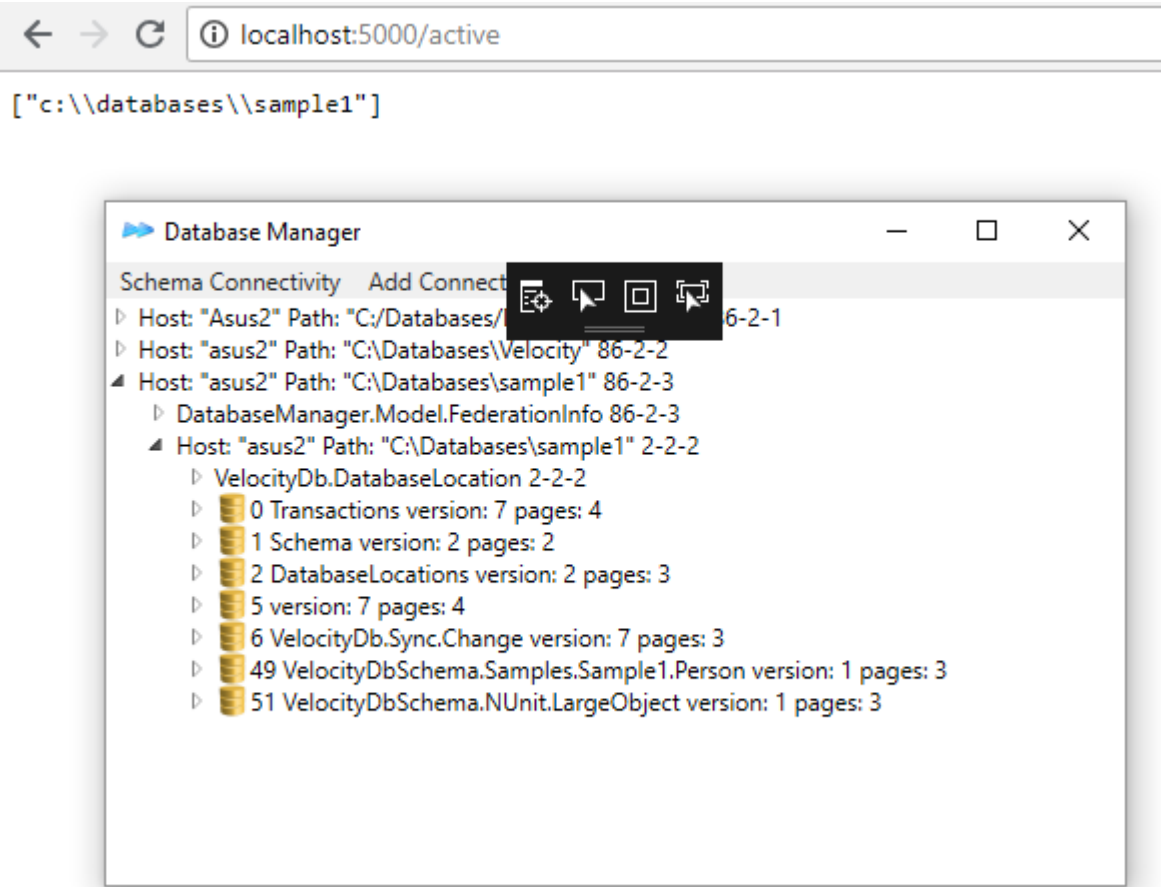


Friday, May 10, 2019

Release History

May 10, 2019	Fixes for export and import to CSV files.	8.2.1
April 16, 2019	Add SessionBase.UpdateObject(IOptimizedPersistable obj, Action codeBlock) and SessionBase.UpdateObject(object obj, Action codeBlock) to facilitate some parallel updates of objects. See test Paralell in ConcurrentUpdaters NUnit test for usage sample. Some fixes to SessionNoServerShared was also made.	8.2
March 30, 2019	Fix ImportFromCSV so that schema doesn't get corrupted. Fix for persist and read of nullable enum objects.	8.1.8
March 26, 2019	Add some checks with exceptions thrown if FieldInfo is missing from a Type field. Usually means the type was modified without calling SessionBase.UpdateClass.	8.1.6
March 23, 2019	Bug fix for reading objects.	8.1.5
March 19, 2019	Fix for Android library.	8.1.4
February 28, 2019	Moving from .net core 2.1 to 2.2. Updating EdgeType.GetEdge with two extra optional parameters. Fixing default SessionBase.BaseDatabasePath for Android. Fix for SessionBase.RelocateDefaultDatabaseLocation	8.1.3
February 15, 2019	Adding VelocityGraph mode to Database Manager	8.1
February 7, 2019	Introducing database replication. See User's Guide and test in Replication.cs.	8
January 6, 2019	LinqPad driver updated to show generic classes and exclude non-relevant (not in schema) types. Fixed VelocityGraph so that it ensures fields loaded for DatabaseManager which uses lazy object field load.	7.7.1
November 8, 2018	Bug fix for BTreeMap that will reduce memory usage. Separated .Net Core projects from .Net 4.5 projects in provided solution. Add Vertex.RelatedVertices. Updated Mac/Linux default SessionBase.BaseDatabasePath to be current directory instead of /Databases. Not sure what is best default in this case?	7.7
September 9, 2018	A concurrency bug fixed. http REST API expanded in VelocityDBCoreServer, see https://velocitydb.com/UserGuide.aspx#_Toc524178339	7.6
September 3, 2018	Fixes for DataMember.StringToType. Preview release of VelocityDB http API via Asp.Net Core 2.1 server combined with regular VelocityDBServer functions all in one. Connect as: localhost:5000/active, for now this alternate server uses port 7032 for VelocityDBServer functions. .NET Core have a mismatch with regular .NET in type names. Specifically noticed so far is mscorlib -> System.Private.CoreLib . To test using this server instead of regular .NET one, set SessionBase.s_serverTcpIpPortNumber = 7032. Some fixes applied for .NET core users, especially when mixed with regular .NET using the same databases. Current http API for now just have one access point showing all server active session database paths. Plan is to expand this so that almost everything imaginable can be done: retrieving object by id, creating new objects, deleting objects, updating objects... What we probably will not support: <ul style="list-style-type: none">- Creating new classes or other Type instances. DLLs containing the application classes will have to be provided to server, so it can work with such object instances. Let us know what you would like to see in this https REST API? We really appreciate all input. The source code will soon be included in our sample solution and on GitHub.	7.5

	 <p>The screenshot shows a web browser window with the address bar containing "localhost:5000/active" and a refresh button. Below the address bar, the text ["c:\\databases\\sample1"] is displayed. In the foreground, a "Database Manager" window is open, showing a tree view of database objects. The tree view includes:</p> <ul style="list-style-type: none"> Host: "Asus2" Path: "C:/Databases/..." 6-2-1 Host: "asus2" Path: "C:\Databases\Velocity" 86-2-2 Host: "asus2" Path: "C:\Databases\sample1" 86-2-3 <ul style="list-style-type: none"> DatabaseManager.Model.FederationInfo 86-2-3 <ul style="list-style-type: none"> Host: "asus2" Path: "C:\Databases\sample1" 2-2-2 <ul style="list-style-type: none"> VelocityDb.DatabaseLocation 2-2-2 <ul style="list-style-type: none"> 0 Transactions version: 7 pages: 4 1 Schema version: 2 pages: 2 2 DatabaseLocations version: 2 pages: 3 5 version: 7 pages: 4 6 VelocityDb.Sync.Change version: 7 pages: 3 49 VelocityDbSchema.Samples.Sample1.Person version: 1 pages: 3 51 VelocityDbSchema.NUnit.LargeObject version: 1 pages: 3 	
August 25, 2018	Make IOptimizedPersistable & OptimizedPersistable JSON serializable friendly by replacing some Properties with getter/setter methods. Page is now GetPage/SetPage, WrappedObject -> GetWrappedObject, Shape -> GetTypeVersion/SetTypeVersion and Session -> GetSession.	7.4
August 19, 2018	Add SessionBase.UpdateObject(IOptimizedPersistable). Improve page and object loading performance in DatabaseManager by using a couple of new lazy load functions added to VelocityDB for Database page iteration (Pages()) and Page objects iteration (ObjectsLazyLoaded). A bug related to multi session use while doing Compact() fixed in SessionNoServer.	7.3.1
August 4, 2018	If a license database is missing when a license check is happening, VelocityDB will copy a license database from your "Downloads" directory if such a file exists and use it for the license check.	7.3
June 8, 2018	Fix build issue of net4/VelocityEngine.dll, it was not re-signed after obfuscation.	7.2.1
May 12, 2018	Updated .net standard 2 version of VelocityDB to include all functionality. All VelocityDB core libraries now named VelocityDb.dll as having multiple assembly names proved infeasible as the name is embedded in type names. We tested switching DatabaseManager, VelocityDBServer and NUnit test suite to use the .net standard DLL instead of regular .NET. It all worked fine but delivering these versions in the installer proved a little difficult so we switched back to use .NET 4.7.1 for	7.2

	these instead. WARNING: If you are currently using VelocityDB with an assembly not named VelocityDb.dll you will see issues upgrading to this new version where its renamed to VelocityDb.dll. You will need to stay with your current version or start over with fresh data. We might be able to help migrating data but it's difficult (we tried it).	
April 18, 2018	Fix for SessionBase.ReplacePersistedType and OfType for a type that isn't yet in database schema.	7.1.7
March 10, 2018	Fix for ISerializable with circular references via List instances used in serialization. Solution is to not embed List object bits in their parent object. Instead give each list an Id like most other objects. Clients with existing data can disable the List with Ids by setting <code>SessionBase.EmbedISerializableLists = true;</code>	7.1.6
March 5, 2018	Fix an issue with ISerializable related to timing of when ISerializable get persisted.	7.1.5
March 3, 2018	Fix incomplete handling of ISerializable	7.1.4
February 17, 2018	Fix SessionBase RelocateDatabaseLocationFor.	7.1.3
February 8, 2018	Fixes to changes made in version 7.1	7.1.2
February 8, 2018	Add support for viewing weak references in Database Manager Schema Connections and added way to register a type as using weak references and to what type(s). Updated User's Guide.	7.1
January 26, 2018	Added Schema Connectivity window to Database Manager and added/updated some supporting functions in VelocityDB.	7.0
December 30, 2017	Added CompressedBitMap to VelocityDBExtensions. We updated the source code to work efficiently with VelocityDB. Stackoverflow uses the original C# code. All sample project sample data files are now consolidated into a single folder named C:\SampleData.	6.12
October 15, 2017	VelocityDB.dll now built with .NET 4.5 due to requiring use of Monitor.IsEntered API not found in prior versions. Fixes for SessionNoServerShared and ServerClientSessionShared to work with updates to pages and indices. A test named ConcurrentUpdatesShared was added for this. Let us know if you have code that still does not work as expected?	6.11
October 1, 2017	Added support for; persisting Types such as List<string[]>, NodaTime type of persisted objects (no fields) display in Database Manager and LinqPad (updated driver).	6.10
September 29, 2017	Added SessionBase.ReplacePersistedType. Updated User's Guide and UpdateClass sample.	6.9
September 21, 2017	Fix for zero sized objects used with ServerClientSession. Improved message for types not found in loaded assemblies.	6.8.2
September 12, 2017	Fix VelocityGraph NuGet package	6.8.1
September 10, 2017	Updated BTreeSet and BTreeMap for reduced memory usage and improved performance when used with large sets of data. Fix for SessionBase.Compact(). Added collection class WeakShortReferenceList and updated WeakReferenceList to allow transient usage. Updated User's Guide. Older versions of VelocityDB will not be able to read the updated BTreeSet and BTreeMap objects used with this release. Previously persisted data using BTree/BTreeMap will be updated to new schema when accessed within an update transaction.	6.8
September 1, 2017	Fixes for ServerClient.Abort() related to initial transaction and new databases. Better error message in License exceptions. DatabaseManager updated with Page.Offset info for each page.	6.7
August 27, 2017	SessionBase.Compact() totally redesigned now much faster. Pages now in page number order after a Compact(). Database cache validation now more efficient with SessionNoServer.	6.6
August 19, 2017	Upgraded for .net core 2.0 and .net standard 2.0	6.5
August 14, 2017	Fix for final delete of deleted databases. Fix of a deadlock in VelocityDBServer when multiple servers are involved.	6.4.2
August 13, 2017	Bug fix to session page offset tracking; make sure to clear updated page offset array at transaction commit. Thanks for test case Harry and Bernard!	6.4.1

August 12, 2017	Make <code>Index</code> attribute implicit when using other index attributes such as <code>UniqueConstraint</code> . Allow setting page and object caching defaults (not persisted setting) for selected databases. Fix bug in setting <code>AutoPlacement</code> object to updated when current transaction id is 0. Fix for memory usage estimates in <code>DataCache.MemoryUsedByThisProcess</code> . (fewer flushes to disk)	6.4
June 28, 2017	Fix persist of empty embedded <code>List<string></code> . Add NUnit test for it (Tony.cs).	6.3.1
June 26, 2017	Added <code>SessionBase.RelocateDefaultDatabaseLocation()</code> and <code>RelocateDatabaseLocationFor(UInt32 dbId, string hostName, string directory)</code> . Using this API is now required whenever moving databases.	6.3
June 24, 2017	Fix for updating default <code>DatabaseLocation</code> after moving databases to a new directory/host. Fixes for build of <code>VelocityDBExtensions</code> project.	6.2.3
June 23, 2017	Don't do string intern by default. Fix schema migration of objects sharing page with other types of objects (not common).	6.2.2
June 22, 2017	Improved inverted index code in <code>TextIndexer</code> and <code>Wikipedia</code> samples.	6.2.1
May 28, 2017	VelocityGraph : Avoid storing duplicate string property values for each property type and a fix for <code>Traverse</code> API.	6.2
April 7, 2017	Fix for subscription changes and a <code>BTreeBase</code> fix. <code>DeletePage</code> fix in server.	6.1.4
March 12, 2017	Updated API documentation (red text due to missing code comments should be gone in most places). <code>VelocityDBCore</code> and <code>VelocityDBStandard</code> now built with Visual studio 2017.	6.1
March 6, 2017	Update code to make strong object reference caching possible without enabling strong reference page caching. Fix for zero size objects.	6.0.10
February 14, 2017	Fix for: <code>BTreeMap.Values</code> , nullable struct display in <code>DatabaseManager</code> , fields typed as interface.	6.0.9
February 12, 2017	Don't disable cache of database 0 with <code>SessionNoServer</code> for better short update transaction performance.	6.0.8
February 6, 2017	Avoid some costly available memory size checks for 64 bit processes in cache management.	6.0.7
January 28, 2017	Fix for type lookup when deserializing <code>ISerializable</code> , increase number of object/page for <code>BTreeByteArray</code> and a few other <code>BTree</code> classes.	6.0.6
January 10, 2017	Fix for deserialization of enum array.	6.0.5
January 9, 2017	Fix for array of enum values. Updated <code>AllSupportedSample</code> class with a sample for this. <code>Database Manager</code> updated to display such arrays with enum value strings.	6.0.4
January 4, 2017	String to Type fix in prior version fixed <code>int[, ,]</code> but broke other cases so make fix more specific to type string containing "[,,"	6.0.3
December 29, 2016	Fix persisted enums data with underlying type as <code>UInt16</code> . Fix for regression of deserializing type string of types like <code>int[, ,]</code> as revealed by <code>AllSupportedSample</code> .	6.0.2
December 22, 2016	Support non <code>IOptimizedPersistable</code> types in <code>LinqPad</code> . Persist non <code>IOptimizedPersistable</code> objects to one database per type as expected. <code>AzureSession</code> extension moved from <code>VelocityDBExtensions</code> to <code>VelocityDBExtensions2</code> due to dependencies requiring a newer .NET version.	6.0.1
December 17, 2016	Replaced <code>IRelation</code> interface with <code>IReferenceTracked</code> and <code>ReferencedTracked</code> . Added a simple <code>Relations</code> sample to show example usage of this API. Moved all exception classes to a separate namespace. Add a .Net Standard 1.6 library to Nuget and download (<code>VelocityDBStandard.dll</code>). Updated driver to <code>LinqPad</code> – added a couple of missing references.	6.0
October 2, 2016	Moved <code>BTree</code> extensions to namespace <code>VelocityDb.Collection.BTree.Extensions</code> . <code>BTreeExtensions</code> . If you are using indexes or doing LINQ queries with <code>BTreeSet</code> , you need to add: <code>using static VelocityDb.Collection.BTree.Extensions.BTreeExtensions</code> to your files that does or else index/ <code>BTreeSet</code> lookups will be slow. The same extensions are also provided in the <code>VelocityDBExtensions</code> library which is included with source code in the download and now also as a NuGet. To use that code instead of the <code>VelocityDB</code> build in extensions do <code>using static</code>	5.8

	VelocityDBExtensions.Extensions.BTree.BTreeExtensions; instead and you will be able to debug the code and hopefully help us improve this tricky expression tree code.	
September 21, 2016	Now supporting System.Transactions.	5.7
September 4, 2016	Adding support for .net core. VelocityDBServer source code added to solution provided in download.	5.6
August 28, 2016	Enable index fields with IOptimizedPersistable. Some fixes for updated DynamicDictionary test. Fixes for DateTime indexing. LoadFields moved from IOptimizedPersistable to SessionBase to simplify interface.	5.5.1
August 20, 2016	Several bugs fixed as revealed by new DynamicDictionary test. DynamicObject base class and dynamic now better supported. Thanks for the test Dean! GeoHashSample updated.	5.5
August 14, 2016	Fix for csv export of fields typed as interface type. Avoid clearing cached objects from weak references when clearing cached objects to save memory use.	5.4.6
August 6, 2016	Add explicit clear of weak reference object cache when clearing potentially out of date objects in a database. Our online issue manager was seen picking up these invalid objects. With this change, this issue is gone.	5.4.5
July 26, 2016	Fixes to AllObjects and OfType, especially Skip.	5.4.4
July 18, 2016	Additional fix for ISerializable. Added option to include/exclude selected Vertex types in Vertex.Traverse	5.4.3
July 17, 2016	Remove unnecessary use of ToArray() in VelocityGraph.	5.4.2
July 15, 2016	Added support for fields of type Type[] and now persisting Type info parameter for ISerializable SerializationInfo	5.4.1
July 10, 2016	Now supporting ISerializable . This was added to support special classes/structs such as those in the NodaTime library.	5.4
July 8, 2016	Fix for null toVertex case in Vertex.Traverse. Now finds all possible paths from a given Vertex within max hops.	5.3.19
June 24, 2016	Properly handle serialization of null fields typed as System.Object or as interface. Introducing ServerClientSessionShared for shared reads from VelocityDBServer with multiple concurrent threads.	5.3.18
June 4, 2016	Updated OptimizedPersistent.Equals and User's Guide	5.3.17
May 20, 2016	Added LinqPad driver source code to sample solution and GitHub. Fix for showing struct in DatabaseManager. Added prototype Sync related stuff to DatabaseManager.	5.3.16
May 3, 2016	Use a weak reference set of shared sessions in VelocityDBServer and add and fix cleanup code.	5.3.15
April 30, 2016	Fix VelocityDBServer to process requests separately for each system database directory.	5.3.14
April 25, 2016	Updates to Azure file stream api code in AzureSession class. More works now. Added AzureFileApi.cs to NUnit tests.	5.3.13
April 22, 2016	Fix for string -> Type conversion. Thanks Marco for the tip!	5.3.12
April 21, 2016	Fix for updating assembly version in persisted type strings. Schema is updated when using a different VelocityDB assembly version compared to version used by prior update transaction. This is unfortunate but currently required because current .Net class loader is using assembly version when resolving a Type. Any better ideas? Fix for handling of closed network streams in VelocityDBServer.	5.3.11
April 17, 2016	Fix for CSV import.	5.3.10
April 14, 2016	Add a missing lock in VelocityDBServer.	5.3.9
April 7, 2016	Enable multiple threads/sessions and processes to add objects, including same type of object, concurrently using ServerClientSession. NUnit test in MultipleUpdaters.cs updating to test this.	5.3.8
April 3, 2016	Add NUnit test to MultipleUpdaters.cs for testing with multiple sessions and multiple threads adding same object type to a couple of databases using ServerClientSession. Fix a couple issues found by this new test. Allow reading previously committed page from VelocityDBServer with pessimistic locking while page is updated by another transaction.	5.3.7
March 29, 2016	Another fix for fields typed as interface.	5.3.6

March 27, 2016	Verified to work with Microsoft Azure cloud file storage. Added prototype AzureSession class (project VelocityDbExtensions). AzureSession is still work in progress but using shared Azure drives directly using "net use" commands works OK.	5.3.5
March 24, 2016	Fix for fields typed as interface.	5.3.4
March 22, 2016	Fix cache management regressions	5.3.3
March 21, 2016	Cleanup: remove a no longer used parameter (disableFlush) of IOptimizedPersistable.Update and Unpersist (some minor code changes may be required). Add OptimizedPersistable.UpdateNonIndexField().	5.3.2
March 17, 2016	Allow updating indexed object without removing and re-adding (at page flush) object from indices (if object uses indices) when updating a NON indexed field. Update non indexed field by calling SessionBase.UpdateObject instead of OptimizedPersistable.Update() . See Indexes sample class Customer.StreetAddress for sample usage. Started on code using Microsoft Sync Framework but we can't find any samples showing how to do it and API documentation is not good enough. Anyone? Code is in Sync.cs part of VelocityDbExtensions project.	5.3.1
March 6, 2016	Added a prototype for a Sync framework and a SyncTest in NUnitTest. Schema for this may change so don't deploy it yet. Please provide feedback! The Sync code is currently open src in VelocityDbExtensions project.	5.3
February 28, 2016	Fix a case of persist failure with non IOptimizedPersistable class field.	5.2.1
February 27, 2016	Cleanup namespaces in VelocityDB (some minor code changes may be required). Add IRelation interface and Relation classes as an attempt to solve some referential integrity issues. Sample usage in FileFolder NUnit test. Please provide feedback!	5.2
February 22, 2016	Move some code in OptimizedPersistable used by Database Manager to VelocityDBExtensions DLL.	5.1
February 17, 2016	Fix a thread safety issues for BTree/BTreeMap reads when used with SessionNoServerShared	5.0.40
February 16, 2016	Improved thread safety for SessionNoServerShared.	5.0.39
February 14, 2016	Fix for memory usage check and added WeakReferenceList.ClearAndUnpersistContainedObjects	5.0.38
February 11, 2016	Permit query by interface such as session.AllObjects< OptimizedPersistable >()	5.0.37
February 4, 2016	Updated (rev 2) notification at commit for updated, new and deleted objects.	5.0.36
February 2, 2016	Nuget not properly build in prior version. Xamarin stuff (or something is deleting some files at Visual Studio startup)	5.0.35
February 1, 2016	Prototype (preliminary) SessionBase API for notifications at commit for newly persisted objects, deleted objects and updated objects. Sample usage in NUnit FileFolderTest. Fix for WeakReferenceList.CopyTo(T[] array, int arrayIndex)	5.0.34
January 25, 2016	Fix for fixed persist of fixed size strings (fields declared withStringLength attribute such as [StringLength(250)]). Added SessionBase.Transaction auto rollback by Dispose() helper class for use with SessionBase.BeginUpdate.	5.0.33
January 24, 2016	Make SessionPool IDisposable. Enable browsing of Licenses in DatabaseManager. Add schema (start of it) and NUnit test for storing folders and files in VelocityDB.	5.0.32
January 17, 2016	Fix an object open usage with short id (UInt32). Fix for a database cache issue in VelocityDBServer.	5.0.31
January 12, 2016	A cache management fix.	5.0.30
January 9, 2016	Remove SessionBase destructor (finalizer) and add IDisposable to DatabaseCache. Disable shared schema between ServerClientSession instances due to some issues.	5.0.29
January 7, 2016	Remove Database destructor (finalizer) and add IDisposable to Page. Use String.Empty instead of ""	5.0.28
January 4, 2016	Lazy loading now does not initially load arrays (except strings and nullable arrays). Removed parameter for BeginUpdate() related to checking for boot dir changes. Moving system database directory no longer requires a manual DatabaseLocation update.	5.0.27
January 1, 2016	More fixes in ServerClientSession. Lazy load of file bytes in issue manager file attachments. Updated default for DataCache.MaximumMemoryUse, is is now 80% of RAM available.	5.0.26

December 31, 2015	Fix for ServerClientSession.	5.0.25
December 30, 2015	Updates to Database Dispose() and enable sorting databases by name in DatabaseManager. Fix for recovery.	5.0.24
December 21, 2015	Upgrade all Nugets in sample solution.	5.0.23
December 21, 2015	Enable drill down into all known object references in Database Manager.	5.0.22
December 17, 2015	Disable recovery check in BeginRead() by default for performance reasons. Updated DatabaseManager and User's Guide. File extension .odb now configured by installer. An .odb file now opens in Database Manager and has an assigned icon. Database Manager will ask you to specify class file(s) (if not already done) for selected .odb file.	5.0.21
December 13, 2015	Added support for Nullable enum. Updates to WeakReferenceList. Added AutoClaim (6 projects) sample.	5.0.20
December 7, 2015	Update VelocityGraph NuGet to use latest VelocityDB NuGet. Updated User's Guide.	5.0.19
December 4, 2015	Added check for missing Type definition of a subclass when doing AllObjects.	5.0.18
December 2, 2015	Added WeakReferenceList to collections. Updated VelocityGraph (note : incompatible changes) to use weak references only. Added static property SessionBase.ClearAllCachedObjectsWhenDetectingUpdatedDatabase (by default is true). Added SessionBase api with template type parameter for object Open.	5.0.17
November 29, 2015	Fix some caching issues. Update VelocityWeb sample to use recommended way to persist objects and to use SessionPool for update transactions and a shared SessionNoServerShared for read only transactions when using IssueDatabases.	5.0.16
November 24, 2015	Change load of schema so that assembly version is handled better when used with software like cAlgo. Add an upper limit to how many UpdateTransaction objects are logged in Database 0. Set limit by property SessionBase.MaxUpdateTransactionLogHistory. Update Vertex.Traverse api to allow a null target Vertex and find all paths up to a certain max number of hops.	5.0.15
November 20, 2015	Fix for adding objects to pages in certain conditions. Improved Type name strings in DatabaseManager object browser. Some updated sample code.	5.0.14
November 18, 2015	Fix for Vertex.Traverse and GraphJson export (source and target Vertex were reversed)	5.0.13
November 16, 2015	Updated Vertex.Traverse API. Fix for Encryption in DatabaseManager. Updated GraphJson support for VelocityGraph exports.	5.0.12
November 10, 2015	Fix for DateTimeKind with persisted nullable DateTime.	5.0.11
November 9, 2015	Fix a BTreeLeaf persist issue. Updated TriangleCounter sample.	5.0.10
November 8, 2015	Fixes to DatingRecommendations sample and added a Vertex.GetNumberOfEdges overload. Added SessionBase.BTreeAddFastTransientBatchSize	5.0.9
November 3, 2015	Improve DatabaseManager in use with concurrent updates. Fix some concurrency issues with SessionNoServer. Fix for databases copy (full backup).	5.0.8
November 2, 2015	Persist DateTimeKind with each persisted DateTime.	5.0.7
November 1, 2015	Add caching of caching of typeof. Might improve performance slightly.	5.0.6
October 26, 2015	Updates to SessionBase.RestoreFrom and Database Manager.	5.0.5
October 25, 2015	Fix for schema cache in server session when system databases deleted in file system or by non-server client session.	5.0.4
October 23, 2015	Fixes for SessionBase.RestoreFrom. Added backup & restore and new/edit DatabaseLocation support to Database Manager	5.0.3
October 20, 2015	Fixes for Database Manager plus added capabilities: Copy (backup) and Validate. No changes to VelocityDB.dll	5.0.2
October 19, 2015	Fix for AllObjects iteration when deleted slots exist on a page. Added start of Database Manager which eventually will have functionality for database backup, backup, restore and browsing. Multiple "federations" can be browsed concurrently and selected federations are persisted in a special DatabaseManager federation (system database location).	5.0.1
October 13, 2015	Added support for Universal Windows applications - .NET application compiled into machine code the same way as C++	5.0

	applications. Using Trace instead of Console for trace output.	
October 6, 2015	Fix some locking/cache issues with SessionNoServer.	4.14.2
October 5, 2015	Fix some file lock timeout issues.	4.14.1
September 28, 2015	Moved csv import/export from VelocityDB.dll to open src in project VelocityDBExtensions. Fix for a BTree unpersist issue.	4.14
September 24, 2015	A fix for OfType/AllObjects Count(). Add a SessionPool class.	4.13
September 19, 2015	GeoHash added to VelocityDBExtension and a sample code project using it called GeoHashSample	4.12
September 16, 2015	Use VelocityGraph nuget package in samples. Fix VelocityGraph nuget dependency to Frontenac.Blueprints	4.11.4
September 16, 2015	Add missing packages.config files in sample solution.	4.11.3
September 15, 2015	Added QuickStart VelocityGraph project. Fixed a bug with including generic subclasses using SessionBase.OfType/AllObjects.	4.11.2
September 10, 2015	Added updated NUnitTests project to VelocityDB.sln, configured all projects to use VelocityDB nuget instead of reference to download installed VelocityDB.dll. Put VelocityGraph samples in separate solution folder. The VelocityDB.sln and all its project source code is now maintained on GitHub at https://github.com/VelocityDB/VelocityDB	4.11.1
September 8, 2015	Added support using VelocityDB for Asp.Net Identity and a sample web site using it. Both projects in download provided solution (VelocityDB.sln). Added SessionBase.InUpdateTransaction.	4.11
September 6, 2015	Updated Sample4 with fuzzy string matching sample code.	4.10.2
September 1, 2015	Upgrade iOS project to Xamarin Unified (64-bit support). VelocityDBMono fix for Linux.	4.10.1
August 29, 2015	Add QuickStart, WebApi and WebApiClient projects to VelocityDB.sln (part of product download/installation). The WebApi is a starting point for a full RESTful interface to VelocityDB and VelocityGraph. Please suggest to us what you would like us to add to this http interface.	4.10
August 23, 2015	Fix VelocityDBMono.dll to run correctly on Linux with QuickStart sample	4.9.5
August 18, 2015	Merged ImdbImport sample into KevinBaconNumbers sample and updated it to use simple Persist (as recommended). Some updates to User's Guide.	4.9.4
August 16, 2015	Moved RTree source code from VelocityDB.dll to open source in VelocityDBExtensions project which is part of the sample solution VelocityDB.sln. We currently don't have an RTree sample project. Help us improve the RTree code or create a sample!	4.9.3
August 5, 2015	VelocityGraph update: Add missing parenthesis when bit shifting for GetEdges and other function – found by visualizing graph with Alchemy.	4.9.2
July 31, 2015	Upgrade to use Frontenac Blueprints 2.4 including support for GraphJson. Fix a string persist issue related to import of GraphJson graph.	4.9.1
July 18, 2015	Improved performance by eliminating some lookups. Updated Wikipedia sample. Fix for database dispose in server sessions.	4.9
July 15, 2015	Updated TextIndexer sample.	4.8.1
July 11, 2015	Des encryption fix. Fixes to samples. Json export and import support added in separate extensions DLL, VelocityDBExtensions.dll. Added JsonExportImport sample.	4.8
July 5, 2015	Eliminate compiler warnings from sample project build.	4.7.2
June 26, 2015	Updated all samples to build with changed compression API.	4.7.1
June 13, 2015	Added lz4 as a new page compression option. PageInfo now also contains info about size of page bytes uncompressed. Fix for optimistic locking with SessionNoserver. Fixes for recovery. Warning: older versions of VelocityDB can NOT read data created with this release. Backup your data before using this release.	4.7

June 7, 2015	Move all thread locking code from SessionBase & SessionNoServer to SessionNoServerShared. Improves performance of SessionNoServer.	4.6.5
June 3, 2015	Fix a regression where threading support code assumed session object created in same thread as main thread running it.	4.6.4
May 31, 2015	Added GraphJson export/import functions for VelocityGraph class Graph. Use with graph visualizer Alchemy java script tool. Updated SupplierTracking sample end by exporting and importing graph.	4.6.3
May 30, 2015	Slightly improved performance of worker threads in SessionNoServerShared, especially with pessimistic locking. Cleanup in server locking code and avoid exception in server when attempting to create a database that already exist.	4.6.2
May 24, 2015	Reduce writer threads in SessionNoServerShared to 1. Fix for regression in delete page and fix for null ptr in AllObjects Count().	4.6.1
May 16, 2015	Added option (on by default) to do object byte encoding and page writes in separate worker threads, one per database (only for SessionNoServerShared at this time). Indexing in a separate thread now also requires using SessionNoServerShared. SessionBase.Compact() improved and now should be run outside scope of a transaction.	4.6
April 27, 2015	Performance boost, added option (on by default) to handle add to indexes in a separate worker thread (only for SessionNoServer at this time).	4.5
April 24, 2015	Performance boost, optimize byte encoding for empty arrays and lists.	4.4.7
April 22, 2015	Performance boost by caching indexing attributes in database schema instead of querying field for indexing attributes every time.	4.4.6
April 18, 2015	Fix regression in use of SessionBase. BaseDatabasePath and fix AutoIncrement to work with non UInt64 fields like a UInt32 field. Updated Indexes sample.	4.4.5
April 16, 2015	Speed up SessionNoServer constructor (avoid some Dns host name checks). EdgeType/VertexType unpersist fixes.	4.4.4
April 15, 2015	Enabled more Linq queries to use fast path by using direct index BTree lookups instead of iterating index. An EdgeType unpersist fix.	4.4.3
April 10, 2015	Add check for invalid use of edge/vertex property type.	4.4.2
April 9, 2015	Enable search for vertices in sub classes by optional extra parameter when using VertexType.GetPropertyVertex and VertexType.GetPropertyVertices.	4.4.1
March 28, 2015	Enabled more Linq queries to use fast path by using direct BTree lookups instead of iterating index. Added SessionBase.TraceIndexUsage as a way to debug incorrect index usage. Added MaximumMemoryUse to VelocityDBServer.exe.config file. Made SessionBase.Persist less recursive by limiting persist to object itself and its direct fields (not anything reachable from object).	4.4
March 16, 2015	Fixed regression in LINQ driver (due to session constructor api change)	4.3.1
March 15, 2015	Revert change to call session.FlushUpdates() in session.Index<>. It is too inefficient – see index section in User’s Guide. Added OptimizedPersistent.Write(). Removed unused parameter “readOnly” from SessionNoServer constructor.	4.3
March 12, 2015	Fix for fixed size objects when in a nested class hierarchy when leaf one has a zero size. Update indices before when fetching corresponding BTree.	4.2.7
March 4, 2015	Fix for AutoIncrement when used in a base class. Updated LINQPad driver for new LinqPad version.	4.2.6
February 27, 2015	Placement fix, use IOptimizedPersistable.PlacementDatabaseNumber if overridden to return something other than Placement.DefaultPlacementDatabaseNumber. Fix for an [UseOidShort] issue.	4.2.5
February 12, 2015	Guard against setting property values to null (instead remove Edge/Vertex property value) and add IOptimizedPersistable as	4.2.4

	a property type choice. Handle errors better on server side. Now removing objects from indices when deleting a database.	
February 5, 2015	Fix a bug in BTree code causing null pointer exceptions. Thanks Tony for the test case!	4.2.3
February 3, 2015	Enable object caching by default (Make CacheEnum.Yes default in session constructors) and release all cached objects when a cached database is not up to date (updated by another session). This can have a dramatic impact on performance. One client saw elapsed time go from 15 minutes to 4 s.	4.2.2
January 28, 2015	Add support for nullable struct types such as Guid?	4.2.1
January 24, 2015	Fix of Traverse in VelocityGraph and added a new sample for VelocityGraph named SupplierTracking.	4.2
January 18, 2015	Fix for comparison of directory path for DatabaseLocation (with / or \ as separators and case insensitive)	4.1.1
January 18, 2015	Added SessionBase.Compact() (not yet implemented for ServerClientSession). Improved handling of Unpersist – now updates AutoPlacement so that slot is reused when a new objects of the same type is persisted. Updated license check.	4.1
January 2, 2015	Added support for updating database locations after moving databases to a new directory/host with ServerClientSession.	4.0.1
December 28, 2014	Added untested versions of VelocityDB.dll for iOS and Android. The names of these added DLLs are: VelocityDBiOS.dll and VelocityDBAndroid.dll. They are part of the normal download, placed in installation folder. Do license check only when updating database schema. This enables deployment without a license database (4.odb). Updated User's Guide describing this for deployments.	4.0
December 21, 2014	Allow changing tcp/ip port number used with VelocityDBServer.	3.24
December 15, 2014	Fix a bug in Database page bytes allocation. Size of Database page bytes was sometimes calculated incorrectly.	3.23.1
December 14, 2014	Added optional bool signalError = true parameter to SessionBase.NewDatabase	3.23.0
December 7, 2014	Add SessionBase option for using String.Intern to save memory. Refresh some cached FileInfo to better handle deleted files.	3.22.0
December 3, 2014	Fix for SessionBase.UpdateClass and updated UpdateClass sample application.	3.21.3
December 2, 2014	Fixed and updated FacebookGraph (VelocityGraph) sample application	3.21.2
November 19, 2014	Fix transaction log persist bug	3.21.1
November 16, 2014	Add SessionBase.OfType and Database.OfType as a non-template class complement to AllObjects<T>.	3.21
November 10, 2014	Add more log data for server logging and some rearrangement of acquire read/update locks in server.	3.20.1
November 9, 2014	Add BTreeBase.GetKeyId and BTreeMapBase.GetValueId as requested.	3.20
November 9, 2014	Added a LINQPad driver for VelocityDB. It still has some limitations but see it as a preview. Feedback is welcome. Added information about LINQPad driver to User's Guide .	3.19
November 2, 2014	Add an edge list verification function optional parameter to Vertex.Traverse. Fill in some missing VelocityGraph api documentation.	3.18.2
November 2, 2014	Fix Vertex.Traverse bug and add more optional parameters.	3.18.1
October 26, 2014	Added more optional parameters to Vertex.Traverse	3.18
October 25, 2014	Fix for VelocityGraph Traverse and added a parameter includedVertices.	3.17.2
October 19, 2014	Fix for [AutoIncrement] field attribute.	3.17.1
October 18, 2014	Added [AutoIncrement] field attribute.	3.17
October 11, 2014	For ServerClientSession, only invalidate pages updated by another transaction instead of invalidation an entire Database. Updated session constructor parameter.	3.16
October 8, 2014	Fix directory compare and transaction id increment in VelocityDBServer.	3.15.5
October 4, 2014	Make strong object references cache default in SessionNoServer constructor and switch to weak object reference cache when memory available is low, switch back to strong references when more memory is available.	3.15.4

October 2, 2014	Enable strong object reference cache with ServerClientSession constructor.	3.15.3
September 28, 2014	Replace a Queue with a RingBuffer (as in Ndisruptor) in server request management.	3.15.2
September 27, 2014	Use BufferedStream for most network writes to reduce networks packets.	3.15.1
September 25, 2014	Use BufferedStream for network writes to reduce networks packets.	3.15
September 21, 2014	Fix a recursive issue in recovery	3.14.1
September 20, 2014	Add databasePerType parameter to SessionBase.AllObjects for faster lookups.	3.14
September 14, 2014	TcpClient.NoDelay = true to speed up client/server communication.	3.13.1
September 13, 2014	Optimize IEnumerable Skip by overriding extension method for AllObjects.	3.13
September 1, 2014	Add lock around writes to server activity log.	3.12.2
August 31, 2014	Fix persist of nullable ValueType.	3.12.1
August 30, 2014	Add option to log activity in VelocityDBServer and fix some error management in VelocityDBServer.	3.12
August 17, 2014	Store full dns host name persistently in DatabaseLocations but try short host name if full host name lookup fails	3.11.1
August 16, 2014	Add support for Guid type for BTree comparison byte array (for faster index lookups/inserts).	3.11
August 14, 2014	Fix server page request bug.	3.10.5
August 13, 2014	Allow null strings in CompareByField.	3.10.4
August 11, 2014	Fix for an infinite loop in SessionBase FlushUpdates and correct comparison array compare of double bytes with negative values. Use comparison array data for indices of fixed length value type fields.	3.10.3
August 9, 2014	Propagate updated optimistic locking setting to slave servers and make pages uncompressed by default (for improved speed)	3.10.2
August 6, 2014	Fixes for Indexing feature with ServerClientSession.	3.10.1
August 3, 2014	Added SessionBase.CopyAllDatabasesTo and added option to include subclasses in AllObjects iterators.	3.10
July 29, 2014	Persists fixes and recursive flush fix.	3.9.4
July 26, 2014	Update to session UpdateClass and AllObjects<T> iterators. UpdateClass sample updated as well.	3.9.3
July 22, 2014	Update to Wikipedia sample application and a couple of related fixes.	3.9.2
July 20, 2014	Fix a persist issue as revealed by inverted index creation in Wikipedia sample application.	3.9.1
July 12, 2014	Updated HighAvailability sample, allow custom directory for VelocityDB installation.	3.9
July 10, 2014	Fix some internal Persist issues of referenced objects.	3.8.8
July 9, 2014	Improve performance of AllObjects iterator and add AllObjects<T> Count().	3.8.7
July 7, 2014	Fix an issue with primitive members only loading.	3.8.6
July 6, 2014	Fixed and updated HighAvailability sample, now always loading primitive members of an object even when using graph depth restriction or lazy load. (as a user requested)	3.8.5
June 29, 2014	SessionBase ExportToCSV and ImportFromCSV now completed (we hope).	3.8.4
June 25, 2014	Fix for an array persist bug.	3.8.3
June 22, 2014	1000x faster connection to VelocityDbServer (work around MS bug). http://social.msdn.microsoft.com/Forums/en-US/4f99a5b3-7f35-4eec-9ac0-c71c309ebd39/new-tcpclientserverport-constructor-slow-on-64bit-computers?forum=vbgeneral	3.8.2
June 21, 2014	Fix for a schema cache bug.	3.8.1
June 14, 2014	Fix some host name related issues, fix regression in HighAvailability sample and add SessionBase.BaseDatabasePath enabling use of relative paths in session constructors.	3.8

June 8, 2014	Fix support for non IOptimizedPersistable with AllObjects iterator and support for non-embedded string persist.	3.7.10
June 7, 2014	Fix build warnings and add a missing unpersist for a VelocityGraph BTreeSet.	3.7.9
May 26, 2014	Some object placement fixes as revealed by regression in Wikipedia sample application.	3.7.8
May 15, 2014	VelocityGraph now using Frontenac.Blueprints 2.9.10	3.7.7
May 4, 2014	ExportToCSV and ImportFromCSV are almost done. Fix for in memory usage with cached schema in SeverClientSession.	3.7.6
April 30, 2014	Resolve a DLL namespace conflict	3.7.5
April 29, 2014	Cache database schema with ServerClientSession to improve performance (~5x)	3.7.4
April 26, 2014	Use <code>Dns.GetHostEntry</code> within try catch and use <code>Dns.GetHostAddresses</code> instead where possible. Enable console output from VelocityDbserver when started from a cmd.exe	3.7.3
April 23, 2014	Try allowing ip number usage as alternative to hostname when dns hostname isn't available.	3.7.2
April 19, 2014	Added a preview (format not finalized) of SessionBase ExportToCSV and ImportFromCSV (not yet fully working). The goal is to be able to use CSV files for data backup and possibly as an aid when making drastic schema changes. Added a possible fix for an issue with use of Monitor in VelocityDbServer which happened on certain installations joined to a domain. Now using Windows Phone 8.1 SDK for VelocityDbWindowsPhone.	3.7.1
April 5, 2014	Added a preview (format not finalized) of SessionBase ExportToCSV. ImportFromCSV is coming a little later. The goal is to be able to use CSV files for data backup and possibly as an aid when making drastic schema changes. Added a fix for external storage in Windows Phone 8 DLL.	3.7
March 29, 2014	Upgraded to use Frontenac.Blueprints 2.3.9 including Frontenac.Gremlinq for queries!	3.6.7
March 27, 2014	Fix for simple persist – handle ShortId members like keyList of SortedMap.	3.6.6
March 26, 2014	VelocityGraph updated to use Frontenac.Blueprints 2.3.8 including Frontenac.Gremlinq for queries!	3.6.5
March 14, 2014	Support field types with IsSecurityTransparent true property value (as in F# sample application). Fix for concurrent use of VelocityDbServer.	3.6.4
March 12, 2014	Better handling of zero sized object types and allow read only browse of databases in a moved directory.	3.6.3
March 11, 2014	Upgraded to use Frontenac.Blueprints 2.3.7 including Frontenac.Gremlinq for queries!	3.6.2
March 10, 2014	Guard against recursive session FlushUpdates() and reduce flushing. Updated SCHEMA (incompatible – sorry!) for VelocityGraph to improve property performance. More placement into type separate database files. VelocityGraph now updated to use Frontenac.Blueprints 2.3.6 including Frontenac.Gremlinq for queries!	3.6.1
March 8, 2014	Add support for descending sort in CompareByField. Updated SCHEMA (incompatible – sorry!) for VelocityGraph to improve edge performance.	3.6
March 4, 2014	Update to VelocityGraph to use Frontenac.Blueprints 2.3.5, add IndexInTree to BTree iterators and a BTree extension method to override the default LINQ IEnumerable.Count() one.	3.5.2
March 2, 2014	Bug fix for InternalErrorException : Adding weak reference page that ...	3.5.1
February 15, 2014	Added support for common multi-dimensional arrays such as int[,] and improved array performance.	3.5
February 8, 2014	Faster comparison of strings with CompareByField comparison array data. Added option to have Vertex ids be globally unique within a graph as alternative to maintaining Vertex ids per VertexType. This is an incompatible schema change for VelocityGraph. It's an option requested by a VelocityGraph client.	3.4.3
January 27, 2014	Fix for concurrent page flush with ServerClientSession and a nasty obfuscation issue.	3.4.2
January 24, 2014	Fix Vertex id allocation.	3.4.1
January 15, 2014	Added support for persisting List of enum and NonIndexed Graph properties. Object oriented graph api added: VertexType	3.4

	and EdgeType can now have a base type. Some api related to this may still be missing. Let us know if you notice anything missing. This change is incompatible with prior version of VelocityGraph (VertexType and edgeType schema change)	
January 13, 2014	Fix for AddFast when used with ServerClientSession	3.3.2
January 1, 2014	Iterator cleanup, more api implemented for VelocityGraph (incompatible change made)	3.3.1
December 26, 2013	Added support for Windows 8 Phone (beta) – see User’s Guide. Added sample DatabaseLocking and WindowsPhoneSDCard	3.3
December 21, 2013	Fixes for deadlock detection and lock wait in server. User selectable id (UInt32) when creating Vertex in VelocityGraph.	3.2.2
December 8, 2013	Fixes for [Use0idShort] cases.	3.2.1
December 7, 2013	BTree and placement fixes and performance improvements. Fix of BTree string compare. Added BTree API AddFast (for large ingest/batch/bulk cases).	3.2
November 30, 2013	Added VelocityWeb sample web application using VelocityDB. It includes managing users and IssueTracker. User’s Guide updated to describe installation to a hosting server.	3.1
November 24, 2013	Fixes in session classes.	3.0.4
November 23, 2013	Fixes to recovery check in VelocityDbServer.	3.0.3
November 20, 2013	Fixes to backup in VelocityDbServer.	3.0.2
November 17, 2013	Improved exception handling in VelocityDBServer.	3.0.1
November 13, 2013	ServerClientSession now has an added in memory only option. High Availability sample updated to show off in memory option. Many fixes to VelocityDbServer and ServerClientSession.	3.0
November 2, 2013	Fix for VertexType.RemoveVertex and BTree fix for delete (collapsing to empty leaf node)	2.16
October 27, 2013	Disable Windows Authentication by default when connecting to a VelocityDBServer. See User’s Guide how to enable it. Some updated to VelocityGraph.	2.15
October 26, 2013	VelocityGraph fix for Vertex.GetVertices. Added VertexType.GetTopNumberOfEdges .	2.14.1
October 25, 2013	Updated FacebookGraph and DatingRecommendations samples. Fixes for VelocityDbServer (removed linger option) and a mutex change. Fixed unidirectional edges.	2.14.0
October 20, 2013	Added more efficient and recyclable Vertex and Edge Ids. Fix for VelocitydbList Contains.	2.13.0
October 13, 2013	Added FacebookGraph VelocityGraph sample application. Updates to VelocityGraph. Added Traverse Vertex to any Vertex.	2.12.0
October 10, 2013	Updated VelocityGraph DatingRecommendations sample.	2.11.7
October 10, 2013	<i>VelocityGraph performance improvements</i>	2.11.6
October 9, 2013	Fix for VelocityGraph Vertex GetNumberOfEdges. Updated VelocityGraph DatingRecommendations sample.	2.11.5
October 6, 2013	Fixes to VelocityGraph and new more advanced sample added named DatingRecommendations	2.11.0
October 5, 2013	Fix to AllObjects session iterator. Fix to persist with large number of transactions.	2.10.1
October 2, 2013	Installer is now code signed. Avoid repeated open of Database 3 looking for indexes when no indexes exist. Use some code generated read functions for some of the build in classes (avoids use of reflection).	2.10.0.0
Sep 22, 2013	Fix for UpdateClass sample. Updated User’s Guide. Preparing for deployment code generation for .NET and unmanaged C++. IOptimizedPersistable updated.	2.9.9.0
Sep 16, 2013	Fixes to sample solution so that sln config AnyCPU isn’t required for a successful build	2.9.8.0
Aug 30, 2013	Add support for index usage in query with multiple field equal test, like <pre>var q = from v in session.Index<TimeArray>(db) where v.MyDate == myDate && v.Sym == sym</pre>	2.9.7.0

	<pre> && v.Stu == stu && v.Analysis == analysis && v.Days == days && v.CheckW == checkw select v.TArray; </pre>	
Aug 24, 2013	Fixes to ElementAt and added error handling for WeakIOptimizedPersistableReference , thanks Tony for test case!	2.9.6.0
Aug 19, 2013	Indexing fixes. Add support for byte[] as key of BTreeSet.	2.9.5.0
Aug 9, 2013	Fix update of indexes when updating indexed fields. Update User's Guide.	2.9.4.0
Aug 8, 2013	Enforce [UniqueConstraint] in indexes	2.9.3.0
Aug 8, 2013	Indexing fixes (mainly related to use of [OnePerDatabase]).	2.9.2.0
Aug 6, 2013	Updated User's Guide, added SessionBase ForceDatabaseCacheValidation, server/client fixes.	2.9.0.0
Aug 5, 2013	Fixes to [OnePerDatabase] indexing and error handling.	2.8.6.0
Aug 5, 2013	Added option to SessionBase for setting default string comparer, default is now StringComparer.Ordinal	2.8.5.0
Aug 1, 2013	Fix for Database.IsNew test, Indexing fixes – no longer using troublesome “dynamic”, faster Database AllObjects ElementAt	2.8.4.0
July 30, 2013	Added <code>public static T ElementAt<T>(this AllObjects<T> source, int index)</code> and going back to using .NET 4.0	2.8.3.0
July 28, 2013	Fix for string to Type loading with signed assemblies and changed assembly version	2.8.2.0
July 27, 2013	VelocityGraph now passes Frontenac.Blueprints tests and is provided as a NuGet	2.8.1.0
July 12, 2013	BTree performance improvements. Renamed WeakReference to WeakIOptimizedPersistableReference . Now using .NET 4.5 instead of .NET 4.0. VelocityGraph now implements the standard graph api called Blueprints. The interface classes used is Frontenac.Blueprints, the project home for that is: https://github.com/Loupi/Frontenac	2.8.0.0
July 10, 2013	Rename a BTree Add function so that it doesn't get mixed up with adding a byte[] value. Fix to persist of zero size objects.	2.7.2.0
June 30, 2013	Fix for internal updates of of NON IOptimizedPersistable types, like BindingList	2.7.1.0
June 29, 2013	Fix for too many enumerator interfaces on OptimizedPersistable (caused trouble with some type of classes). Added a database verifier named Verify. It is a command line application that verifies that all objects and references can be opened without error in a selected set of databases. Fix initialization of database objects LoadedFields property.	2.7.0.0
June 23, 2013	Fix for InTransaction test	2.6.2.0
June 10, 2013	Improve short transaction performance by removing repeated reinitializing of build in schema. Fix in database cache database count.	2.6.1.0
June 9, 2013	Fix possible race condition when connecting to secondary server. Major updates to VelocityGraph	2.6.0.0
June 2, 2013	Client/Server fixes.	2.5.1.0
May 30, 2013	Change event subscription and notification added to ServerClientSession API. EventSubscriber sample application added showing usage of the new api. VelocityGraph updated. VelocityDbServer fixes.	2.5.0.0
May 12, 2013	VelocityGraph library project and sample added to the solution of sample projects. A fix to BTreeMap Persist.	2.4.0.0
May 10, 2013	Added a few missing fixes for DeletePage.	2.3.2.0
May 9, 2013	Fixes to Persist related to regression of SortedObjects sample. Fix to page delete not always releasing weak reference causing free space mess.	2.3.1.0
April 26, 2013	Fixes to VelocityDbServer and added a new High Availability sample application	2.3.0.0
March 30, 2013	Fix a schema initialization issue when a type is unavailable, fix stack overflow case when persisting a very deep graph.	2.2.2.0
March 25, 2013	Fixed persistence of generic Dictionary.	2.2.1.0

March 24, 2013	BTreeMap now implements IDictionary<Key, Value> . User's Guide has been updated. Serialization bug fix.	2.2.0.0
February 24, 2013	Fix a couple of bugs introduced with simplified persistence	2.1.1.0
February 23, 2013	Simplified object persistence, most samples updated to use the simplified persistence. A few bug fixes.	2.1.0.0
February 21, 2013	Indexing fixes.	2.0.4.0
February 14, 2013	Obfuscation issue (too much of it) braking Indexes	2.0.3.0
February 13, 2013	Bug fix for regression of BTree iterator	2.0.2.0
February 9, 2013	More options added for Indexing feature.	2.0.1.0
February 8, 2013	Added Indexing feature. A fix in persisting to class selected database number. Added Indexing and OneDbPerClass sample. BTreeSet now implements ISet<T> interface.	2.0.0.0
January 11, 2013	Fix for persistence of struct, Database caching issues and page delete with server version.	1.9.1.0
November 24, 2012	Added support for automatic use of BTreeSet binary search lookup with LINQ query, updated SortedData sample to illustrate it	1.9.0.0
November 8, 2012	Bug fix for pages with a single type of object and a placement enhancement	1.8.2.0
November 5, 2012	Major updates/fixes for Wikipedia sample, BTree short reference internal placement fixes. HashCodeComparer updated. (not compatible with old version – removed an instance variable). TextIndexer (a “mini” inverted index)updated to work with changes to code in common with Wikipedia sample.	1.8.1.0
October 13, 2012	Added SessionBase.DeleteObject	1.8.0.0
October 10, 2012	Added Page.UnpersistObject for usage when OptimizedPersistable.Unpersist cannot be used. Updated User's Guide	1.7.4.0
October 6, 2012	Fixes for SessionNoServerShared and VelocityDbServer transaction management.	1.7.3.0
October 1, 2012	Fixed a multiple servers scenario bug, fix for updating DatabaseLocations and general code cleanup. Now using Visual Studio 2012 for development but keeping compatibility with vs2010 for now (but not for too long).	1.7.2.0
September 15, 2012	Removing reference to Microsoft.VisualBasic assembly as it appears unavailable for Xamarin.Android. Alternate memory management now used.	1.7.1.0
June 6, 2012	Added spatial index R-tree – more functionality need to be added – let us know what? Fixes for optimistic locking using SessionNoServer .	1.7.0.0
May 31, 2012	Bug fix in FileOpen, shared access Read Write was sometimes only shared access Read. Updated Sample4 with code for deleting persistent objects using Unpersist API.	1.6.1.0
May 13, 2012	Bug fix in fixed size objects on non-shared pages. Added hash code sorting of strings with use of ComparebyField with string fields (faster than regular string comparison). Placement fix override of Persist (such as in BTreeSet)	1.6.0.0
May 5, 2012	Added support for custom comparison of BTree comparison bytes and updated CompareByField to correctly compare string bytes store in BTree comparison byte arrays.	1.5.0.0
April 18, 2012	Fix regression with WeakReferences sample	1.4.11.0
March 30, 2012	Fix a database cache management issue in ServerClientSession	1.4.10.0
March 27, 2012	Fix deserialization of object[] objects	1.4.9.0
March 26, 2012	Fix serialization of object[]	1.4.8.0
March 25, 2012	Fix issues related to server api, WeakReference<T> and fields declared to have type object.	1.4.7.0
March 23, 2012	Fix issue for when an object of type object is stored in a persistent field.	1.4.6.0
March 22, 2012	Fix a recursive issue when attempting to serialize an instance of object, fix for array of WeakReference<T>.	1.4.5.0

March 21, 2012	Fix storage of string and array into field of type object . All data created with prior versions must be discarded (due to modifying a build in schema class)	1.4.4.0
March 20, 2012	Allow class fields to be typed as "object" for storage of any type of object.	1.4.3.0
March 19, 2012	Fix for ServerClientSession OpenAllDatabases, adding FirewallException to installer.	1.4.2.0
March 18, 2012	Fix for: trial license generation, ServerClientSession related to page cache and, avoid corruption by not flushing any full page while persisting members of some object.	1.4.1.0
February 1, 2012	More optimized in memory tracking of very large set of Databases (millions), optimized and parallel inverted index create in Wikipedia sample. Added a first draft of User's Guide to the installation.	1.4.0.0
January 16, 2012	IOptimizedPersistable interface is now working, the AllSupportedSample shows a usage with class PersistenceByInterfaceSnake. The interface contains a little too much, we hope to prune some of it someday.	1.3.0.0
January 9, 2012	Updated VelocityFnBrowser to display BTree keys and values as Oid/OidShort instead of more cryptic UInt64/UInt32. Updates to SessionNoServerShared.	1.2.5.0
January 5, 2012	Updated WorldCities sample by providing a QlikView like interface	1.2.4.0
December 29, 2011	Improvements to TextIndexer and Wikipedia samples. Reduced cost of pageOffset maintenance.	1.2.3.0
December 26, 2011	Faster BTree binary search and byte compare. Started building IOptimizedPersistable interface.	1.2.2.0
December 23, 2011	Fixes and performance improvements to TextIndexer and Wikipedia inverted indexes.	1.2.1.0
December 21, 2011	Fixes for BTree Oid ordering, introducing SessionNoServerShared, better handling of OidShort in schema.	1.2.0.0
December 10, 2011	Fixes for recovery, minor internal schema change. All data created with prior versions must be discarded.	1.1.1.0
December 3, 2011	Enhanced BTree to avoid opening objects when doing Oid compare, adding ImdbImport and KevinBaconNumbers sample applications	1.1.0.0
November 27, 2011	New test found issues related to recovery. These bugs are now fixed.	1.0.7.0
November 23, 2011	Fix for Type change due to assembly version change plus other fixes and some minor api changes.	1.0.6.0
November 17, 2011	Bug fixes, fixed some object placement issues, database reopen issue.	1.0.5.0
November 8, 2011	Bug fixes	1.0.4.0
November 3, 2011	Adding TriangleCounter sample. Server/Client interface fix. Reduced memory usage for large objects.	1.0.3.0
October 26, 2011	Performance boost. 187,000 objects/s to 249,000 objects/s. See Benchmark info on samples page, http://www.velocitydb.com/Compare.aspx WeakReference<T> now requires T object to be persisted before usage in WeakReference<T>.	1.0.2.0
October 22, 2011	No longer require that each OptimizedPersistable subclass have a constructor taking a SessionBase as its only parameter.	1.0.1.0
October 21, 2011	Internal redesign of schema, new design handles changes to persistent object classes. Some new API introduced for it and a new sample added to demonstrate these capabilities, sample name is UpdateClass.	1.0.0.0
October 18, 2011	BTreeMap fixes, Benchmark files updated to use BTreeMap in some cases, TextIndexer GUI fixed. Sample database directories moved to personal documents\VelocityDB\Databases sub folders.	0.9.9.9
October 16, 2011	BTree fixes. Updated TextIndexer sample, added option to index text files from http://www.gutenberg.org/browse/scores/top	0.9.9.8
October 11, 2011	Page caching bug fix. BTree changes to reduce in memory requirements. Adding Wikipedia inverted index sample. TextIndexer sample updated. Added Wikipedia inverted index sample. All data created with prior versions must be discarded.	0.9.9.7
October 5, 2011	Added BTreeMap and BTreeMapOidShort. Updated BTree structure to support key + value. All data created with prior	0.9.9.6

	versions must be discarded. Updated TextIndexer sample.	
September 27, 2011	Added option to name Databases. All data created with prior versions must be discarded.	0.9.9.5
September 26, 2011	Fix in Page object iterator, added samples: Baseball and WorldCities. VelocityDbServer now listens for ipv6 connections instead of ipv4.	0.9.9.4
September 17, 2011	Internal fixes. Lazy loading of FreeSpace (internal), no need to read in read only transaction.	0.9.9.3
September 13, 2011	Added option to preSize database (may avoid file fragmentation). BTree bug fix. Performance boost! Cleanup of some internal classes. All data created with prior versions must be discarded.	0.9.9.2
September 6, 2011	Performance boost! Added support for StringLength attribute (save some space with short strings and possibly make object fixed size). Cleanup of some internal classes. All data created with prior versions must be discarded.	0.9.9.1
August 30, 2011	Fixes to Database Benchmark files, fixes to BTree, added api SessionBase CrossTransactionCache	0.9.9.0
August 28, 2011	Fixes previously unhandled list of nullable ValueType.	0.9.8.0
August 27, 2011	Fixes previously unhandled array of nullable ValueType.	0.9.7.0
August 20, 2011	Fixes previously unhandled nullable ValueType fields. Added AllObjects<T> to SessionBase as aif for Linq usage	0.9.6.0
August 16, 2011	Fixes for backup and restore. Several internal schema changes. Start over with freshly made Databases. Fixes so that VelocityDbServer can handle multiple startup (boot) directories. Fixes for DeleteLocation. Many other various fixes.	0.9.5.0
August 9, 2011	Fixed a bug with Dotfuscator usage. Fixed abort of server transaction appending Database file, now returning aborted page space as free space.	0.9.0.0
August 7, 2011	Removed "VelocityDb.Exception" namespace, exceptions move to namespace "VelocityDb". Added lots of code documentation to the API doc.	0.8.5.0
August 3, 2011	First ever public version	0.8.00