

Changelog

ver. 4.0.9 (2015-11-06)

- bug fixed: XTable: vertical compression with struct types now works again (thanks: Gabbby)

ver. 4.0.8 (2015-09-03)

- changed: SortedSetExtensions now generates the same expressions for .NET and Mono. (thanks to corminlu from GitHub)
- changed: DecimalExtensions now generates the same expressions for .NET and Mono.
- improved: FindNext, FindAfter, FindBefore, FindPrev methods of SortedSet.

The Mono Project has adopted the .NET implementations for some of their libraries.

ver. 4.0.7 (2015-01-13)

- bug fixed: SortedSetExtension.ConstructFromSortedArray finally works under Mono

Even after the bug fix, there are problems under Mono. Simple insert of 10 000 000 records with random keys randomly fails with mono runtime error:

//---

Stacktrace:

```
at <unknown> <0xffffffff>
at (wrapper managed-to-native) object.__icall_wrapper_mono_gc_alloc_obj (intptr,intptr) <IL 0x0000e, 0xffffffff>
at (wrapper alloc) object.AllocSmall (intptr) <IL 0x00047, 0xffffffff>
at STSdb4.Database.XTable`2<long, STSdb4.General.Collections.Tick>.set_Item (long,STSdb4.General.Collections.Tick) [0x00002] in
/home/chris/Desktop/STSdb4/STSdb4/Database/XTable.cs:36
at STSdb4.GettingStarted.Program.Example () [0x0007b] in /home/chris/Desktop/STSdb4/STSdb4.GettingStarted/Program.cs:77
at STSdb4.GettingStarted.Program.Main (string[]) [0x00001] in /home/chris/Desktop/STSdb4/STSdb4.GettingStarted/Program.cs:28
at (wrapper runtime-invoke) <Module>.runtime_invoke_void_object (object,intptr,intptr,intptr) <IL 0x00050, 0xffffffff>
```

Native stacktrace:

```
/usr/bin/mono() [0x4b73d8]
/usr/bin/mono() [0x50f13b]
/usr/bin/mono() [0x423d22]
/lib/x86_64-linux-gnu/libpthread.so.0(+0x10340) [0x7f2a24df9340]
/usr/bin/mono() [0x5fb127]
/usr/bin/mono() [0x5fc703]
/usr/bin/mono() [0x5d86ff]
/usr/bin/mono() [0x5ddefe]
/usr/bin/mono() [0x5de3b9]
/usr/bin/mono() [0x5f4719]
/usr/bin/mono() [0x5f47e4]
[0x41c3f133]
```

Debug info from gdb:

```
Could not attach to process. If your uid matches the uid of the target
process, check the setting of /proc/sys/kernel/yama/ptrace_scope, or try
again as the root user. For more details, see /etc/sysctl.d/10-ptrace.conf
ptrace: Operation not permitted.
No threads.
```

```
=====
Got a SIGSEGV while executing native code. This usually indicates
a fatal error in the mono runtime or one of the native libraries
used by your application.
=====
```

//---

This error in mono runtime is already reported in bugzilla: https://bugzilla.xamarin.com/show_bug.cgi?id=25751
We hope that it will be fixed soon.

ver. 4.0.6 (2014-12-16)

- bug fixed: Persist<T> and DataPersist<T> do not work with AllowNull.All flag. This bug does not affect the default database behaviour (which creates the persist instances with AllowNull.OnlyMembers flag).

- bug fixed: XFile does not Read correctly after open.

- bug fixed: SortedSetExtension.ConstructFromSortedArray finally works under Mono without the slow workaround.
- bug fixed: DecimalExtensions.GetDigits now works under Mono.
- bug fixed: SlotsBuilder.BuildType now works under Mono.
- other: the STSdb4 is built under .NET 4.5 Framework.

The database should now work under Mono.

ver. 4.0.5 (2014-08-05)

- bug fixed: GetDigits() method.

ver. 4.0.4 (2014-06-25)

- incompatible change: Persist<T> and DataPersist<T> serialization logic is changed for structures, GUID and nullable types. In case the user uses such types, STSdb 4.0.4 is not compatible with 4.0.3. If the user uses classes or primitive types, 4.0.4 is compatible with 4.0.3.

- added: TimeSpan support. TimeSpan is now a primitive STSdb 4.0 type.
- improved: GetDigits() methods - this affects all floating-point delta compressions.
- changed: default StorageEngine settings are changed to:

```
public int INTERNAL_NODE_MIN_BRANCHES = 2;
public int INTERNAL_NODE_MAX_BRANCHES = 5; //old value 4;
public int INTERNAL_NODE_MAX_OPERATIONS_IN_ROOT = 8 * 1024;
public int INTERNAL_NODE_MIN_OPERATIONS = 32 * 1024; //old value 64 * 1024;
public int INTERNAL_NODE_MAX_OPERATIONS = 64 * 1024; //old value 128 * 1024;
public int LEAF_NODE_MIN_RECORDS = 8 * 1024; //old value 16 * 1024;
public int LEAF_NODE_MAX_RECORDS = 64 * 1024; //old value 128 * 1024;
```

Default StorageEngine.CacheSize is now 64; //old value 32

These settings make the database a little more ballanced. The overall speed is slightly improved, while the RAM usage is decreased with ~14%.

- bug fixed: Persist<T> and DataPersist<T> do not generate proper code when the type is KeyValuePair or nullable type.
- bug fixed: ValueToString<T> and DataToString<T> does not work in partial To/From conversion mode.

