

DevForce Universal, WinClient, and Silverlight

Release Notes

Version 5.2.9

Table of Contents

Introduction	4
Why Read the Release Notes	4
What's in a Note	4
Read the Installation Guide.....	5
Customer Support.....	6
Version 5	7
Version 5.2.9.....	7
Summary.....	7
New and Improved Product Features.....	7
Version 5.2.8.....	7
Summary.....	7
New and Improved Product Features.....	7
Version 5.2.7.....	7
Summary.....	7
Breaking changes.....	7
New and Improved Product Features.....	8
Learning Resource Updates	8
Defect Repairs.....	8
Known Deficiencies of the Release	9
Version 5.2.6.....	9
Summary.....	9
Breaking changes.....	10
New and Improved Product Features.....	11
Learning Resource Updates	11
Defect Repairs.....	12
Known Issues.....	12
Version 5.2.5.....	12
Summary.....	12
Breaking changes.....	13
New and Improved Product Features.....	13
Learning Resource Updates	13
Defect Repairs.....	13
Known Issues.....	14
Version 5.2.4.....	14
Summary.....	14
Breaking changes.....	14
New and Improved Product Features.....	14
Learning Resource Updates	15
Defect Repairs.....	17
Known Issues.....	18
Version 5.2.3.....	18
Summary.....	18

Breaking changes.....	19
New and Improved Product Features.....	19
Learning Resource Updates	19
Defect Repairs.....	20
Known Issues.....	21
Version 5.2.2.....	21
Summary.....	21
Breaking changes.....	22
New and Improved Product Features.....	22
Learning Resource Updates	24
Defect Repairs.....	26
Known Issues.....	29
Version 5.2.1.....	30
Summary.....	30
Breaking changes.....	30
New and Improved	30
Learning Resources	31
Documentation	32
Defect Repairs.....	32
Appendix. Converting a DevForce Release Candidate Silverlight App To Run Under DevForce 5.2.1 and Silverlight 3 RTW	33
Version 5.1.1.....	39
Summary.....	39
Breaking changes.....	39
New and Improved	41
Learning Resources (formerly "Learning Units").....	43
Documentation	44
Defect Repairs.....	44
Version 5.1.0.....	46
Summary.....	46
Breaking changes.....	46
New and Improved	50
Learning Units	51
Documentation	52
Defect Repairs.....	52
Version 4	54
Version 4.3.0.....	54
Summary.....	54
Breaking changes.....	54
New and Improved	55
Learning Units	57
Documentation	57
Defect Repairs.....	58
Version 4.2.2.....	58
Summary.....	58
Breaking changes.....	58

New and Improved	58
Learning Units	59
Documentation	60
Defect Repairs.....	60
Version 4.2.1.....	61
Summary.....	61
Breaking changes.....	61
New and Improved	62
Learning Units	63
Documentation	64
Defect Repairs.....	64
Known Limitations	65
Version 4.2.0.....	67
Summary.....	67
Breaking changes.....	67
New and Improved	68
Learning Units	71
Documentation	71
Defect Repairs.....	72
Known Limitations	72

Introduction

These notes describe the history of changes to the DevForce product and provide important information before you upgrade. Each DevForce “Feature” release earns its own chapter with more minor “Maintenance” releases listed as topics within the chapter.

We indicate a “Feature” release by changing in the second digit of the DevForce version number (e.g., 3.1 to 3.2); a “Maintenance” release is a change in the third digit (e.g., 3.1.4 to 3.1.5). The fourth digit varies by build and does not typically merit a release note.

Why Read the Release Notes

While the notes may be just a curiosity to the first-time DevForce installer, they are **essential reading** if you are upgrading your DevForce application or thinking about doing so.

The steps you must take every time you upgrade are covered in the **DevForce Installation Guide**. There may be additional, **release-specific upgrade steps**. These you can only learn about here.

Each release topic in these notes calls out the additional steps required to transition from the previous release. There may be none. There may be just a few that can be handled by re-running a DevForce tool and recompiling the application. On rare occasions, you may have to make manual changes to your code.

You will only know if you read these notes. If you are upgrading across several versions, please read the notes for all intermediate versions.

What’s in a Note

Upgrade Advice

We describe what you have to do to migrate to this version.

New Features

We add new features almost every release. New features are often described here in great detail when first introduced so that you can discover and learn about them in one convenient place. We migrate that detail into the **Developers Guide** over time as the “new” becomes the “mainstream”. That’s where most developers will go to learn about programming with DevForce.

Fixes

These notes cover our corrections to defects in the product. You’ll want to check here to confirm we’ve addressed a bug that you submitted.

Backward Compatibility and Signature Obsolescence

We strive to avoid changes that could break your code. Backward compatibility within the DevForce product is a cherished value.

However, we learn from experience and sometimes we find some functionality that needs to be redesigned. The new design may result in method names or signatures that compete with the prior design. We preserve the previous signatures if we can, marking them obsolete so that you can gradually remove them from your code.

IdeaBlade DevForce Release Notes

No one wants to be stuck with poor, unused signatures that clutter your code and confuse the developer. Accordingly, when we mark a method signature obsolete, we often announce that we will remove that signature at a specific future date. Typically, that is six months from the obsolescence date.

We will accommodate any Enterprise Edition customer's need for more time to adapt. Please contact your sales representative when this is an issue and we will make accommodations for you.

Breaking Changes

On rare occasions, we choose to remove the old signature entirely. Any application that depends upon that signature requires some re-coding and recompilation.

Perhaps we determined that there is no way to safely implement the method's intended behavior while that signature exists. Rather than permit your code to proceed in error we'll "force" you to fix it – if you choose to upgrade.

Sometimes we change a method in a dark, infrequently visited corner of the product. We may decide that the risk of deleting or altering a signature that almost no one sees is less than the risk that someone might come along later and depend upon it. We hope we are right and we'll revert quickly if we learn that we were wrong.

In all cases, we will tell you what we did and how to adjust your code to recover from the break. We apologize in advance for this inconvenience; we trust you'll perceive the gain as worth the pain.

Read the Installation Guide

The Installation Guide contains important information about how to install and upgrade your DevForce application to a new version.

Customer Support

For support, you can visit our support forums at www.ideablade.com/forum. Your question may have already been answered there by a DevForce engineer or by someone in our development community.

If you have an active Support Subscription, you can also submit a direct support case via our web site at

<http://www.ideablade.com/CustomerSupportRequestForm.aspx>

When submitting a support case, please try to simplify and isolate your problem as much as possible. If you are submitting a bug report, creating a simple test case using the NorthwindIB database will help us to quickly identify, correct, and test the issue.

IdeaBlade also offers Enterprise Support options as well as a full range of Professional Services to help you build your application. Please contact your sales representative or email info@ideablade.com for more information.

Version 5

Version 5.2.9

Summary

This release encompasses minor feature enhancements.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient Only

- DevForce now supports the Infragistics NetAdvantage 2011.1 control suite for WinForms.
- DevForce now supports the Developers Express 11.1 control suite for WinForms.

Version 5.2.8

Summary

This release encompasses minor feature enhancements.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient Only

- DevForce now supports the Infragistics NetAdvantage 2010.2 and 2010.3 control suites for WinForms.
- DevForce now supports the Developers Express 10.2 control suite for WinForms.

Version 5.2.7

Summary

This release encompasses minor bug fixes and feature enhancements.

Breaking changes

Both breaking changes in this release are security-related:

- The *AspAuthenticatingLoginManager* will now throw a *LoginException* if the supplied credentials fail forms authentication. Previously, an unauthenticated *UserBase* was returned to the client. You may need to modify your code if your application contained special logic to handle these unauthenticated users, or is not expecting a *LoginException*. If you implemented a sub-typed *AspAuthenticationLoginManager* to work around this problem, you may no longer need your custom implementation. [B1144]
- If you are supplying credentials with a *Login* call but have not implemented an *IEntityLoginManager*, when the *AllowAnonymousLogin* flag is true the *GenericPrincipal* / *GenericIdentity* returned by the *Login*

will not have the name supplied with the login. When the *AllowAnonymousLogin* flag is **false** and you have not implemented an *IEntityLoginManager*, a *LoginException* will be thrown. [B1297, B1345]

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient Only

- DevForce now supports the Infragistics NetAdvantage 2010.1 control suite for WinForms.
- DevForce now supports the Developers Express 10.1 control suites for WinForms.

WinClient/Silverlight

- *[Duplicates description under “Breaking Changes”]* The *AspAuthenticatingLoginManager* will now throw a *LoginException* if the supplied credentials fail forms authentication. Previously, an unauthenticated *UserBase* was returned to the client. You may need to modify your code if your application contained special logic to handle these unauthenticated users, or is not expecting a *LoginException*. If you implemented a sub-typed *AspAuthenticationLoginManager* to work around this problem, you may no longer need your custom implementation. [B1144]
- Miscellaneous security improvements. Most changes were internal to the framework, except for the following:
 - The *IdeaBladeConfig* type now contains a property named *AllowAnonymousLogin*, which is set to *true* by default. When set, “anonymous” users may use your application without supplying credentials. When off, only authenticated users may use your application. This is a server-side setting on the *Ideablade.configuration* element:

XML	<ideablade.configuration version="5.00" allowAnonymousLogin="false">
-----	--

- *[Duplicates description under “Breaking Changes”]* If you are supplying credentials with a Login call but have not implemented an *IEntityLoginManager*, when the *AllowAnonymousLogin* flag is true the *GenericPrincipal* / *GenericIdentity* returned by the Login will not have the name supplied with the login. When the *AllowAnonymousLogin* flag is **false** and you have not implemented an *IEntityLoginManager*, a *LoginException* will be thrown. [B1297, B1345]

Learning Resource Updates

None.

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Fixed a problem where null reference parameters to a StoredProcQuery caused an *ArgumentNullException* to be thrown. [B1422]
- When using the Silverlight project template with both DF2009 and DF2010 installed on the same machine the web project added to the solution was for DF2010. [B1425]
- Fix for problems using Include operator with FirstOrDefault and FirstOrNullEntity queries. [B1291]
- Fixed problems serializing DateTimeOffset columns. [B1338]

IdeaBlade DevForce Release Notes

- The shortcut to the DevForce Trace Viewer tool has been restored to the Windows Start Menu for DevForce. [B1357]
- Fixed a problem with temporary ID fixup which could occur when restoring an EntityCacheState. [B1387]

WinClient Only

- The UI BindingManager designers for WinForms, which previously did not work in Visual Studio 2010, now do work.

Silverlight-Only

- None.

Known Deficiencies of the Release

- The ToolBox Installer does not install DevForce controls into the VS2010 ToolBox. To install the DevForce controls, see the paragraphs on “Toolbox Installer fails” in the Troubleshooting section of the *DevForce Installation Guide*.

Version 5.2.6

Summary

This version of DevForce 2009 supports *Visual Studio* versions 2008 and 2010. Either or both versions of *Visual Studio* can be installed on the target machine.

Please note! DevForce 2009 supports .NET version 3.5, not 4.0. **You must set the project that includes your DevForce domain model to target .NET version 3.5. (See instructions below).**

Other projects in your solution may target .NET version 4.0. However, whereas .NET permits a .NET 4.0 assembly to call a .NET 3.5 assembly, it does not permit the converse. Therefore your .NET 4.0 assemblies can use your DevForce domain model, but nothing in your domain model can call into a .NET 4.0 assembly.

DevForce 2010, due for release in early April 2010, will support .NET 4.0, including its new version of the Entity Framework. It will require your domain model to be compiled using .NET 4.0.

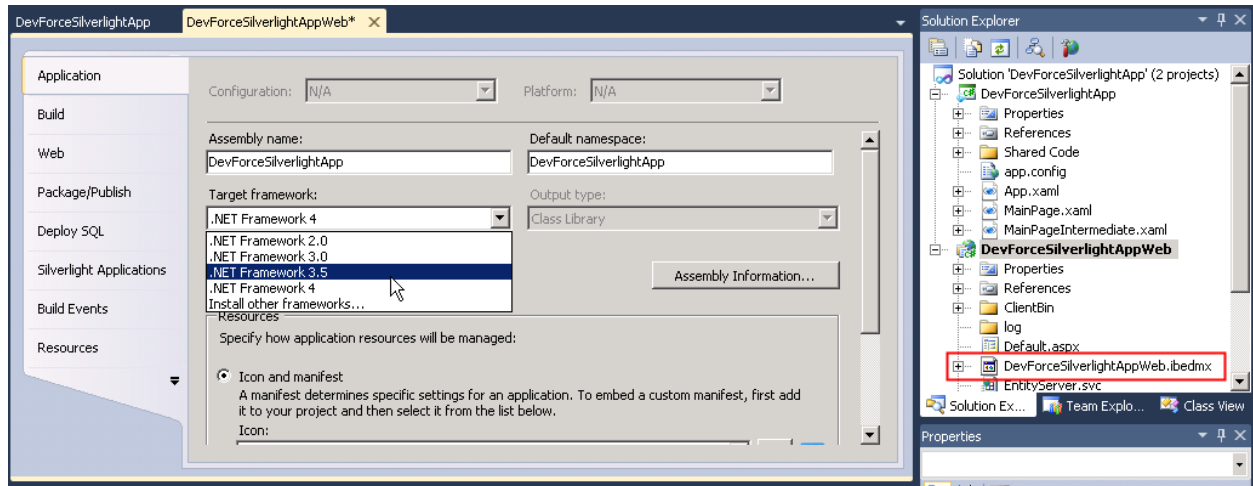
Setting the Target Version of .NET for a Assembly

To control the version of .NET with which a project will be compiled into an assembly, select the project node in the Visual Studio Solution Explorer and right-click. From the shortcut menu, select *Properties*. From that point, the path to the setting depends upon whether your project is in C# or VB:

C#

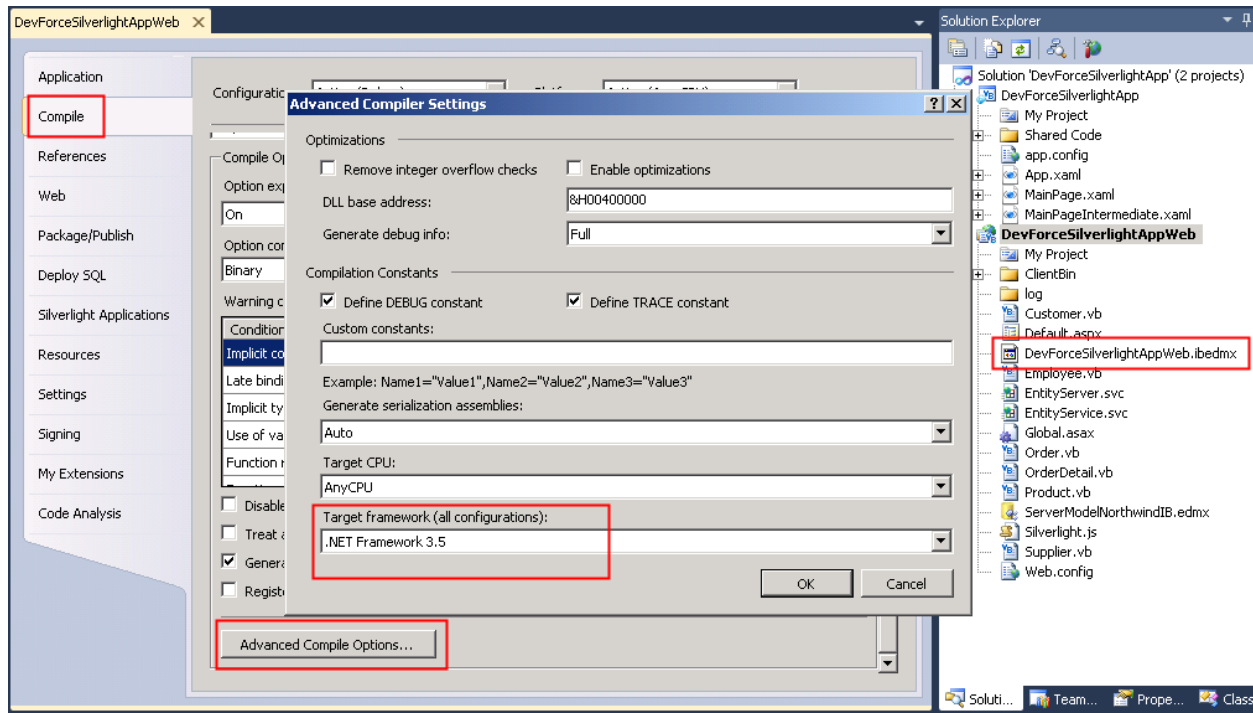
In the Properties dialog window, find the *Assembly* tab. On that tab, set the target version of .NET using the *Target framework* ComboBox:

IdeaBlade DevForce Release Notes



VB

In the Properties dialog window, find the *Compile* tab. On that tab, click the <Advanced Compile Options...> button. In the resulting *Advanced Compiler Settings* dialog, set the target version of .NET using the *Target framework* ComboBox:



Breaking changes

There are no actual breaking changes of which we're aware, but if using Visual Studio 2010 you do need to keep in mind, as mentioned in the Summary for this release, that the project containing your DevForce domain model must target .NET 3.5, and not .NET 4.0.

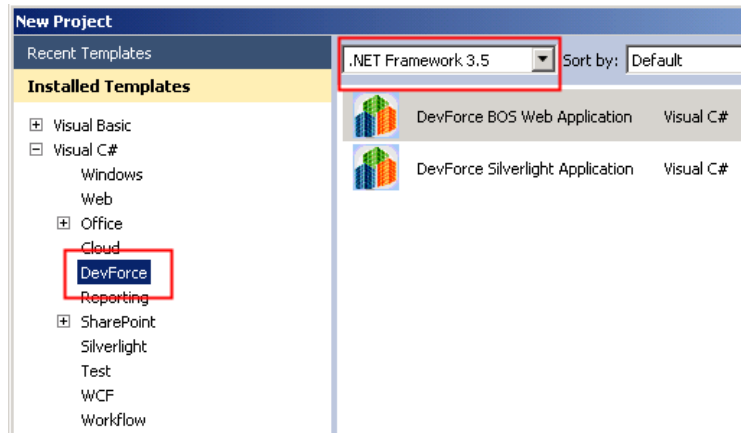
Learning Resources are Maintained in Visual Studio 2008

The code solutions in the Learning Resources are maintained using Visual Studio 2008, so in order to work with them in Visual Studio 2010 you will have to allow Visual Studio to convert the project. When it asks you whether to target .NET 4.0 in the projects in the solution being converted, you should say no, and have the conversion utility leave .NET 3.5 as the target framework. Be aware that if you've previously set your conversion default to target 4.0, you won't get prompted: Visual Studio will just re-target the projects to .NET 4.0 without asking. You will then need to reset their target Framework to 3.5 after the fact. (Follow the instructions provided in the Summary above.)

DevForce Project Templates Are Only Available When Targetting .NET 3.5

When creating a new project in Visual Studio, the DevForce Project Templates for a Silverlight Application or BOS Web Application are only available when you specify the .NET 3.5 framework, as shown at right.

You will not see any DevForce templates if the new project dialog is targetting the .NET 4.0 Framework.



New and Improved Product Features

The new feature for this release is support for Visual Studio 2010.

Learning Resource Updates

The following enhancements were made to the DevForce Learning Resources:

Base Apps

- Three new Visual Studio solutions were added which elaborate upon the completed solution to the Four Simple Steps tutorial:
 - The first converts the *MainPageIntermediate* UserControl into a *MainPageView* UserControl with an associated *MainPageViewModel* class.
 - The second adds a Repository class to centralize data persistence operations and permit their abstraction out of the ViewModel.
 - The third adds unit testing based on the Silverlight 3 Test Harness.

You can find these solutions in the LearningResources in the folder
`\030_BaseApps\SilverlightApps\Samples\200SLV_FourSimpleStepsMVVW.`

Business Object Persistence

- Added a Silverlight sample, *AsyncParallelTask*, which demonstrates the use of the *AsyncParallelTask* method to submit queries in batches for parallel execution and a consolidated result.

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Object Mapper save in VS2010 fails on a machine that has never had VS2008 installed [#B1432]
- Templates don't install when receiving "Error 1606.Could not access network location \VisualBasic" [#B1326]
- IdeaBlade assemblies not seen in Add Reference dialog for .NET on VS2010 [#B1288]
- Behavior and messaging improved in circumstance where VS2008 Silverlight 3 Tools are not found [#B1306]
- Need CLR4 uninstall for VS2010. [#B1245]
- Object Mapper now displays an error message when an EDMX generated by Entity Framework 4.0 is added to the domain model. This will occur when an EDMX is contained within a project which targets .NET 4.0. [#B1249]
- Fixed an issue in which substitution macros in the generated build event commands were not used for some directory structures. The build event is used to copy the server model assembly and any metadata artifact files to a target folder and is auto-generated by DevForce. [#B1289]
- Fixed a problem in which the fixup of temporary Ids was not performed after an EntityManager.SaveChanges(IEnumerable) call. The problem occurred only when abstract base types were involved and a "partial" save was performed. [#B1300]
- Fixed a StackOverflowException which occurred when navigating within the Object Mapper in VS2010. [#B1304]
- Provided a means of overriding the default values DevForce assigns to non-nullable properties when entities are created. [#B1310]
- Object Mapper: Set the initial directory for the file open dialog displayed when adding an EDMX to the domain model. [#B1314]

WinClient Only

- None.

Silverlight-Only

- None.

Known Issues

None.

Version 5.2.5

Summary

This version of DevForce includes a minor bug fix and some documentation updates.

Note: This version does NOT work with Visual Studio 2010. If you are interested in a beta of DevForce for Visual Studio 2010, please email info@ideablade.com.

Breaking changes

None.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient Only

- Support added for Developer Express DExperience 9.3 controls (for WinForms) [F1263]

Learning Resource Updates

The following enhancements were made to the DevForce Learning Resources:

Intro to DevForce

The walk-through to illustrate building a Visual Studio solution using DevForce was updated as follows:

- Various minor errors were corrected.
- A Visual Basic version was provided for all code snippets
- The section on building a model from a web service was removed. (The capability remains in the product, but we simply decided to remove this section from the walk-through, as being of limited interest to most readers.)

In addition, code solutions were added (in both C# and VB), for user reference, to reflect the completed work described in the walk-through. These solutions encompass the creation of the entity data models, DevForce domain models, console and WinForm user interfaces, and test project. They do not include the Silverlight material; readers are referred to the Four Simple Steps tutorial for an example of Silverlight development with DevForce.

Business Object Persistence

- In the *Business Object Persistence* topic document, a new PredicateBuilder example was added “Example: Simulate an In() Clause Condition on a Distantly Related Entity”. The example was also added to the corresponding C# and VB code solutions in the *_TopicDocumentSnippets* folder.
- *Business Object Persistence* document. The following changes were made:
 - ✓ Completed testing and rework of all snippets, C# and VB.
 - ✓ Added a section “Implementing Server-Side Life-Cycle Event Handlers”, which provides instructions for implementing handlers for the ServerFetching, ServerFetched, ServerSaving, and ServerSaved events.

Validation

- A code snippet was added to illustrate the use of the *PropertyNameToDisplayNameTranslator*

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Fixed an issue where a temporary ID was not assigned when the entity had an abstract base class for some types of inheritance. The lack of a temporary ID caused creation and save failures. [B1284]

IdeaBlade DevForce Release Notes

- Install previously did not check for the (required) presence of Microsoft® Silverlight™ 3 Tools for Visual Studio 2008 SP1. It now does.
- An issue was corrected in which, if DevForce was installed without the Learning Resources initially, it was not possible to add them later without reinstalling DevForce. [B1246]
- Corrected a problem under Windows Vista in which an unnecessary (and empty) LearningResources folder appeared underneath the main DevForce installation folder. [B1254]
- Corrected a problem in which certain xml files needed for DevForce-related Intellisense were not installed. [B1260]

WinClient Only

- Fixed a problem causing a NullReferenceException during Dispose processing of the Infragistics 9.2 UltraGrid for WinForms. [B1255]
- Corrected an installation problem in which DevForce BOS project templates were not installed properly when installing DevForce WinClient Enterprise. [B1253]

Silverlight-Only

- None.

Known Issues

None.

Version 5.2.4

Summary

This version fixes a significant performance issue when loading a large number of items that have a single parent. It also incorporates enhancements to the EntityWrapper class, and a number of defect repairs.

Many updates and additions to the documentation and other Learning Resources are included as well. All changes are detailed in the material below.

Note: This version does NOT work with Visual Studio 2010. If you are interested in a beta of DevForce for Visual Studio 2010, please email info@ideablade.com.

Breaking changes

All Products

None.

Silverlight-Only

None.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient and Silverlight

- **Enhancements to the EntityWrapper class.** Extension of *IEditableObject* methods (*BeginEdit*, *EndEdit*, *CancelEdit*) has been enabled on the *EntityWrapper*; and the following new protected virtual methods have been added: *BeginEditCore*, *EndEditCore* and *CancelEditCore*. DevForce entities (and *EntityWrapper*-based classes) can now participate in *IEditableObject* because these “core” methods can be overridden in custom implementations. [F1236]

Silverlight-Only

- None

Learning Resource Updates

The following enhancements were made to the DevForce Learning Resources:

Base Applications

- A code sample, **Simple Master-Detail WinForm**, was added. This application, which is a port from the DevForce Classic code sample entitled "Doing More in Code", uses a single main WinForm to display Employee data in loose controls such as TextBoxes and ComboBoxes. Related Orders and their line items (OrderDetails) are displayed in .NET DataGrids, as are Customers related to the Employee through common Orders. (The Employee acted as a Sales Representative on an Order placed by the Customer.)
- From the topic document “DevForce Silverlight Apps”, an obsolete section on Deployment Silverlight apps to IIS was removed. Said section has been replaced by more comprehensive documentation in the Deployment topic document.

Business Object Mapping

- A code sample, “Inheritance Models”, was added to illustrate the use of different types of inheritance models (Table Per Hierarchy, Table Per Type, and Table Per Concrete Type) with SQL Server, the Entity Framework, and DevForce.

Business Object Persistence

- **_TopicDocumentSnippets.** Two new code solutions, one in C# and one in VB, were added in a new folder directly under the main *Business Object Persistence* topic folder. These solutions contain all code snippets in the *Business Object Persistence.pdf* topic document, with enough context to make the solution compilable and the snippets runnable. In the process many snippets were reworked to bring them up to date or make them more illustrative; and all snippets were translated to VB.

Besides giving you, the DevForce developer, a handy way to exercise and experiment with the material, the solutions will also make it easier for us at IdeaBlade to discover when any snippet has been obsoleted by changes to DevForce so that we can update the documentation in a timely manner. Our plan is to create similar code solutions for all topics, a project at which we will chip away, as time permits.

- **InvokeServerMethodAsync to Preload Cache.** A code sample was added (under the sub-topic *Remote Service Methods*) which illustrates how to call a server method asynchronously to preload the cache with an arbitrary collection of data. The solution is a Silverlight one, but the code to call and use the results of the remote service method has no Silverlight-specific aspects and will work just as well in another type of application. Solution in C#; key code elements also translated into VB in a separate text file.

IdeaBlade DevForce Release Notes

- **TraceViewer.** Two code samples were added. One (available in both C# and VB) illustrates how to embed the WinForms and WPF TraceViewers in your own app. A second (currently available only in C#) illustrates the use of a Silverlight UserControl to monitor client-side trace messages.
- **Checkpointing.** A Console App code sample was added to illustrate the basics of checkpointing, a technique for creating in-cache “transactions”.

The discussion of checkpointing (in the section “Multi-Level Undo with Checkpoints”) was moved from the WinForms UI topic document (*WinForm User Interfaces.pdf*, in the *User Interfaces* topic) to the *Business Object Persistence_Advanced.pdf* document.

- **Business Object Persistence** document. The following changes were made:
 - ✓ Reworking, enhancement, and translation to VB of all code snippets in the chapter as discussed above under the entry **_ TopicDocumentSnippets**.
 - ✓ Reworked the discussion of the PredicateBuilder, clarifying obscure points, adding a new example, and providing VB versions of all code snippets.
 - ✓ **Navigation Properties in Silverlight.** A new section was added following the existing discussion on *Parent-Child Navigation Properties*, to discuss and provide examples for using navigation properties asynchronously, as required by Silverlight, and as also available in non-Silverlight environments.
 - ✓ The discussion of use of the TraceViewer was updated to address the WPFTraceViewer as well as the WinForms TraceViewer; expanded to include a discussion of using the TraceViewer in stand-alone rather than embedded mode; and expanded to address the viewing of trace messages in Silverlight solutions. Code solutions (already mentioned) were added to contain the demos.
 - ✓ Remove an obsolete boxed discussion of “ParentReference”.
 - ✓ Fixed various minor infelicities
 - ✓ Removed discussion of the obsolete ThrowExceptionOnSaveFailure option.
 - ✓ Corrected references to “StoredProcRdbQuery” to the correct “StoredProcQuery”.
 - ✓ Removed obsolete discussion of Cache Search with Stored Procedure Queries
 - ✓ Corrected discussions and code snippets referencing DataRowVersion to use the EntityProperty.GetValue() method.
 - ✓ Revised the discussion on asynchronous communication with the BOS, and asynchronous queries specifically to bring it up to date.
 - ✓ Added a discussion of the EntityManager.CacheStateManager and its facilities for capturing, saving, and restoring the EntityCacheState.
 - ✓ Transplanted, to the *Business Object Mapping – Advanced* topic document, the discussion “DevForce and Data Sources - Digging Deeper” from the *Business Object Mapping* topic document. Renamed same to DevForce Data Sources – Deep Dive”.
- **Business Object Persistence – Advanced** document. The following changes were made:
 - ✓ Transplanted the section “DevForce and Data Sources - Digging Deeper” from the *Business Object Mapping* topic document, renaming it to “DevForce Data Sources – Deep Dive”.
- **AddingDeleting Code Sample – WinForm Version - VB.** Upgraded the VB translation to incorporate all facilities and refinement present in the C# version.

User Interfaces

- Moved the discussion of checkpointing (“Multi-Level Undo with Checkpoints”) from the WinForms UI topic document (*WinForm User Interfaces.pdf*) to the *Business Object Persistence_Advanced.pdf* document in the Business Object Persistence topic.
- Corrected references to the (DevForce Classic) PersistenceManager in *WinForm User Interfaces.pdf*. These were changed to *EntityManager*. Code snippets that included obsolete code (appropriate only for *DevForce Classic*) were updated.

IdeaBlade DevForce Release Notes

Validation

- Reworked the first 11 pages of the *Validation Through Verification* topic document to update screen shots and code samples, and to make the topic more rapidly accessible.
- Fixed a problem with saving changes in the 200WNF WinForm code sample. The solution now correctly permits saving.
- Added a new section, “Adding Attributed Verifiers to Generated Properties” to the discussion of property value verifiers in the *Validation Through Verification* topic document. Also corrected an obsolete code snippet showing the code for a FirstName property generated by the DevForce Object Mapper.
- Replaced the previous WinForm code sample with one based on the reworked VB version of the AddingDeleting solution (which you can find in the folder LearningResources\040_BusObjPersistence\AddingDeleting\Samples\200WNF\CodeVB).

The new sample, unlike the one replaced, includes *all* of the same verification samples facilities as the Silverlight and WPF code samples, including the cross-object verifier defined in the Order class. The latter was missing in the previous version.

- Added a code solution under Code Snippets designed to contain verification-related “living snippets”. Each method in Program.cs illustrates a particular verification-related technique, and is labeled with XML comments to identify the use case. Initial snippets:
 - ✓ *VerifyScalarNavigationProperty()* - Run verification on a scalar navigation property
 - ✓ *SetPropertyNameTranslator()* - Use the PropertyNameToDisplayNameTranslator property on the VerifierEngine to assign a delegate to translate any property name in any way you wish.
 - ✓ *OrderAnInstanceVerification()* - How to order an instance verification

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Previously DevForce was unable to perform ID fixup on C's primary key under the following conditions:
 - A,B,C were related
 - A.Id was autogenerated
 - B had a (1-0..1) relation to A wherein B's primary key was the foreign key to A.Id
 - C had (1-0..1) relation to B wherein C's primary key was the foreign key to B.Id

This problem has been corrected. [B1139]

- A problem was corrected in which DefaultValue attributes that involve Nullable datatypes would fail during EntityManager.CreateEntity calls. [B1188]
- Performance when loading a large number of items having association to a single parent has been greatly improved. [B1206]

IdeaBlade DevForce Release Notes

- *ToQuery(this IEnumerable)* previously threw an *ArgumentNullException* when the list was empty. It now throws an *ArgumentOutOfRangeException* exception with a better explanatory message. [B1191]
- For any executable project in a solution that references a DevForce domain model, DevForce uses a Build event handler to copy Entity Framework “artifact” files and the server assembly to that project’s disk folder at build time. Previously, in certain situations, this handler used an incorrect path. It now uses the *ConfigurationName* macro to specify the target directory, which corrects the problem. [B1087]
- The *EntityManager* copy constructor will now automatically set the *IsConnected* property to *true* in most situations, avoiding lengthy connect processing when not needed. [B1207]
- The *EntitiesWithErrors* property of the *EntityManagerSaveException* now correctly contains the entity in error. Previously, in some situations, the collection was empty. Note that because saves are transactional, when an error occurs then collection will contain only a single entity. [B1215]
- DevForce now allows null parameter values to a *StoredProcQuery*. [B1228]
- The *RequiredAttribute* is no longer incorrectly added to entity properties in generated code when all verification options have been disabled. [B1234]
- Fixed a *NullReferenceException* occurring when a web service method returned an array of items. [B1236]
- Deleting an entity in a many-to-many relationship now correctly cascades the delete to the “link table” also when the related data has been loaded into cache. Note that *this affects only many-to-many relations in which the linking entity has no payload (and is therefore abstracted out of the conceptual model)*. The *EmployeeTerritoryNoPayload* table in NorthwindIB is an example of a payload-free linking table. (It facilitates a many-to-many relationship between the *Employee* and *Territory* tables.) [B1241]

WinClient Only

- None

Silverlight-Only

- Because it was found to have performance issues, the *ThreadPool* is no longer used for asynchronous operations within DevForce Silverlight. [B1210]

Known Issues

None.

Version 5.2.3

Summary

This is a maintenance release of DevForce, with minor improvements and bug fixes.

Other new features and fixes are documented below.

Breaking changes

All Products

None.

Silverlight-Only

None.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient and Silverlight

- The Object Mapper now uses T4 templates to auto-generate code files. This should have no impact on developers in this release, but in a coming release we will provide the ability to override and customize these templates. [F1217]
- The Developer Express version 9.2 control suite (specifically, 9.2.4.0) for Winform apps is now supported. [F1208]
- The Infragistics NetAdvantage 9.2 control suite (specifically, 9.2.20092.1003) for Winform apps is now supported. [F1209]

Silverlight-Only

- Reduced “XML bloat” in messages sent from the server to the client to reduce message size. This change was also made to the EntityCacheState when saving to a stream. [F1205]
- During a build, we now ensure that the *app.config* is marked as an embedded resource in the Silverlight application project. A better explanatory message is also now generated at runtime if an *app.config* is not found. [F1227]

Learning Resource Updates

The following enhancements were made to the DevForce Learning Resources:

Base Applications

- The section, “Silverlight IIS Deployment Steps”, in *DevForce Silverlight Apps*, was moved to the Deployment topic document.
- The section, “Troubleshooting”, in *DevForce Silverlight Apps*, was moved to the Troubleshooting topic document and renamed to Silverlight Troubleshooting”.
- Added a SilverlightConsole app to the Code Samples. This is a DevForce Silverlight app with a very bare bones user interface, somewhat similar in appearance and functionality to the console window of a .NET Console Application. It is intended as a base for quick and dirty mockups of limited and targeted functionality, such as retrieving, modifying, or saving data using the various DevForce asynchronous methods for those operations.

Business Object Persistence

- The Business Object Persistence topic document was divided into two documents: one providing the basics and addressing more advanced topics.

Deployment

- The Deployment topic document underwent extensive revision and updating:
 - It was reorganized to make clearer just what material you need to concern yourself with for a given type of application.
 - A section, “Silverlight IIS Deployment Steps” was moved from the *DevForce Silverlight App* topic document to the *Deployment* topic document. The section was then extensively revised. There are now separate step-by-step guides for II6 and II7.
 - A discussion of probing in DevForce was added. The discussion includes detailed information about where to list probe assemblies in the *app.config* and *web.config* files according to the assemblies’ content.

Troubleshooting

- A section, “Silverlight Troubleshooting” was added, having been moved from the document *DevForce Silverlight Apps*, where it was named simply “Troubleshooting”.

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Documentation was corrected for the CloneCore() method of *Entity*, *EntityWrapper*, and *EntityGroup*. [B1135]
- DevForce now throws an exception in *EntityManager.ImportEntities()* if an entity in the import list is detached. [B1136]
- The <configSections> element for the ideablade.configuration is now written to the correct location of the config file. Previously, it was written to the wrong location when added to an existing configuration file, causing the project not to compile. [B1138]
- DevForce now allows for associations where both sides have a 0..1 to 0..1 relationship. [B1142]
- DevForce now guards against Visual Studio crashes when the Object Mapper opens. [B1148]
- The getters and setters of entity properties can now be marked as non-public without causing an exception or unexpected behavior. [B1155]
- A projection with FirstOrDefault which returns no items now returns the expected default value. [B1174]
- A change to a scalar navigation property now raises the PropertyChanged event for the navigation property (it previously raised the event for only the underlying foreign key property). [B1179]
- We removed informational log messages indicating when a probed type is not found, since these messages were often confused with errors. You will no longer see messages about *IEntityServerFetching*, *IEntityServerFetched*, *IEntityServerSaving* and *IEntityServerSaved* probing unless you have implemented these interface and they are discovered during probing. [B1181]

IdeaBlade DevForce Release Notes

- In solutions with multiple entity models, EDMX files in sub-folders (for example a Data folder in a web project) are now supported. Previously an exception was thrown during generation of the *app.config* file. [B1189]
- Probing for the *IConcurrencyStrategy* interface now searches key-specific probe assemblies before looking at top-level probe assemblies. [B1196]
- We improved exception message text and trace logging for errors occurring during initialization. [B1198]
- A manifest was added to DeploymentTester.exe and other tools [B1081]
- Under Windows Vista and Windows 7, the Learning Resources (code samples, tutorials, and etc.), are now installed in the *Users* folder rather than under c:/program files where they may not be accessible depending upon user's security privileges. Under Windows XP, the Learning Resources continue to be installed under the Program Files folder. [B1140] [B1197]
- To avoid performance problems, only EXEs (and not DLLs) are now Authenticode signed [B1159]
- Corrected a bad error message about Silverlight3 Beta [B1169]
- Corrected failure of automatic uninstall on 64-bit operating systems [B1161]
- InstallShield version upgraded from 2009 to 2010 [B1162]
- The VB version of "Adding and Deleting" for WinForms now correctly saves changes to the database [B1163]
- Corrected problem in which a major upgrade install failed to delete all files in Learning Resources folder [B1184]
- Install process now picks up the correct version of NorthwindIB. [B1187]

WinClient Only

- None

Silverlight-Only

- Instances of *EntityCacheState* can now be returned to a Silverlight application from an invoked server method. [B1199]

Known Issues

None.

Version 5.2.2

Summary

This is a maintenance release of DevForce, with minor improvements and bug fixes. Perhaps the most significant fix was to enable immediate execution queries having an Include clause properly to fetch the related "included" entities.

IdeaBlade DevForce Release Notes

Immediate execution queries are those involving First, FirstOrDefault, Last, LastOrDefault, ElementAt, and ElementAtOrDefault.

Other new features and fixes are documented below.

Breaking changes

All Products

DevForce Upgraded to the Latest WPF Toolkit.

The Object Mapper and WPF Trace Viewer have been updated with the “WPF Toolkit – June 2009 Release”. Prior to this, they used the March 2009 release. Installing this version of DevForce will automatically remove the previous Toolkit assemblies from the GAC and replace them with the newer ones. If you developed any controls based on the March 2009 Toolkit, you may need to reinstall it. [F1171]

Silverlight-Only

The ObjectDataSource and EntityQueryPagedCollectionView have been moved from the IdeaBlade.EntityModel.SL assembly to the new IdeaBlade.Windows.SL assembly. If you use either of these classes, you will need to add a reference to this assembly and a *using* or *Imports* statement for the IdeaBlade.Windows namespace.

New and Improved Product Features

New features are categorized by the DevForce versions to which they apply:

WinClient and Silverlight

- XML serialization via either the DataContractSerializer or the .NET DataContractSerializer for entities is now supported. See the Object Persistence overview document (section “XML Serialization of Business Objects”) in the Learning Resources for details. [F664]
- Default values from the EDMX data model are now applied to newly created entities. Default values are applied by reading the [DefaultValue] attribute on a property. This attribute is now automatically generated based on the EDMX data model, but can also be applied manually to either custom properties or to [MetadataType] extended properties. [F1148]
- We added a new *IdeaBlade.Linq.PredicateDescription*. This is documented in the Object Persistence topic overview document in the Learning Resources. [F1172]
- To the PredicateBuilder, we added a new *PredicateBuilder.Not* method; and also support for the PredicateDescription class mentioned immediately above. [F1169]
- We modified the *EntityKey.ToQuery* and *EntityKeyList ToQuery* implementations to return an *EntityQuery<T>* that is composable using standard LINQ. This is potentially a breaking change because the old signatures returned an *EntityKeyQuery* instead of an *EntityQuery*. The old behavior is still available by calling the new *ToKeyQuery()* method on both the *EntityKey* and the *EntityKeyList* types. The *ToKeyQuery()* variants will be more performant (in the case of a *QueryStrategy* of Normal where the key might already be in the cache), but less composable than the *ToQuery ()* variants. [F1170]
- The EntityAspect class now supports property change notification via *INotifyPropertyChanged*. The following properties are tracked: *EntityState*, *EntityKey*, *EntityVersion*, *IsChanged*, *HasErrors*, and *SavingErrorMessage*. [F1182]

IdeaBlade DevForce Release Notes

- Setting a property to its current value no longer marks the entity as Modified. You can still use *EntityAspect.SetModified()*. [F1183]
- A new *EntityManager.AttachEntity()* method has been provided. It can be used to attach (add) a detached entity to an existing EntityManager without changing its key. The entity is added to the EntityManager in an “unmodified” state.

The method will throw an exception if the entity is NOT in a detached state before being attached, or if the same key already exists in the EntityManager. (You can use the *ImportEntity()* method for the latter; the operation will be subject to the specified MergeStrategy.)

More detail is available in this post by Ward Bell on the IdeaBlade Support Forum:

http://www.ideablade.com/forum/forum_posts.asp?TID=1439&PID=5227#5227

[F1197]

- The Internal method *EntityMetadataStore.GetEntityTypes()* is now public and can be used to get the list of all entity types available within an application.[F1198]
- **Support the DiscoverableMemberAttribute for invoked server methods.** You can now mark the target method of an InvokeServerMethod call with either the DiscoverableMember or AllowRPC attribute.

```
C# [DiscoverableMember(DiscoverableMemberMode.AllowRSM)]
    public static Object GetNumberOfOrders(IPrincipal principal,
        EntityManager entityManager, params Object[] args) {
        return GetOrderCount(entityManager, args);
    }

    [AllowRpc]
    public static Object GetNumberOfOrders(IPrincipal principal,
        EntityManager entityManager, params Object[] args) {
        return GetOrderCount(entityManager, args);
    }
```

[F1151]

- Additional enhancements were made to the *ObjectDataSource* and *EntityQueryPagedCollectionView*. The *ObjectDataSource* control can be used to define declaratively the data source for bound controls. The *EntityQueryPagedCollectionView*, which is used by the *ObjectDataSource* and can also be used standalone, provides a collection of entities using asynchronous execution of an *EntityQuery* for paged results. These are both considered beta features and are subject to change. Samples of both are provided in the Learning Resources. Enhancements are as follows:
 - Grouping and filtering are now supported.
 - SortDescriptions were replaced by SortDescriptors to allow control binding.
 - Visual states are now supported for templating.
 - Additional properties *IsBusy*, *HasChanges*, *IsLoadingData*, *IsSavingChanges*.
 - LoadSize may now differ from PageSize, to control “lookahead” caching of pages.

[F1162]

- Support was added for Windows 7 [F1202]

Silverlight-Only

- Tracing is now supported in Silverlight applications. (Tracing has always been supported in WinClient and web applications, which by default write all trace messages to a log file.) You must add a “trace subscriber” in order to capture trace messages. A sample TraceViewer user control is provided in the Learning Resources. [F1163]
- Implicit login is now performed if an async query, save, or InvokeServerMethod call is made and the EntityManager is not logged in. [F1160]

Learning Resource Updates

The following enhancements were made to the DevForce Learning Resources:


- An HTML menu system was added to permit the resources to be documented and categorized more helpfully, and to facilitate productive browsing through them. This menu system is linked from the installation splash page and from the Windows Start menu for Devforce.


DevForce Learning Resources


The resources in this menu are organized by topic. You'll find the complete DevForce Developers Guide embedded into the topics as individual chapter/topic files. However, if you prefer to have the Developers Guide as a single document, you can find that here: [DevForce Developers Guide](#).


The Learning Resources are always a work in progress. We've organized them by topic first, then by type of resource (document, webinar, tutorial, code sample, etc.), and last by technical level, UI type, and code language. In this way, we hope you'll be able to find a resource that can help you, even if it's not a perfect match for the app you're building, or the language you're working in, or your technical background with DevForce. We'll keep adding new examples and adapting existing examples to additional platforms and languages with each new release of DevForce.

If you need additional help, you can find (or get) answers to many questions on the [DevForce Support Forum](#). Licensed users may also submit direct support requests via the web using the [Technical Support Request](#) form.

[Introduction to DevForce](#)
An overview of the DevForce framework, with videos and documents to get you started

[Learning Pathways](#)
Paths through the Learning Resources to different destinations

[Business Object Mapping](#)
Mapping back-end data sources to business objects

[Business Object Persistence](#)
Retrieving and saving business data

- Several new code solutions and snippets were added.
- Numerous additional code solutions previously available only in C# were translated to Visual Basic.
- Snippets and miscellaneous files previously located in the Samples folder under C:\Program Files\IdeaBlade DevForce were relocated into the Learning Resources folder hierarchy where they are more logically categorized and can be better documented.

Base Applications

- To the document *DevForce Silverlight Apps*, a new section was added, “Silverlight IIS Deployment Steps”.
- Two Silverlight apps were renamed, more accurately and/or descriptively. “FirstSilverlightApp” was renamed to “SimpleSilverlightApp”; “SilverlightNextSteps” was renamed to “SeparateModelProjects”.
- The article associated with the *Four Simple Steps* solution was reformatted to permit code to be copied therefrom and pasted into a Visual Studio class file without loss of end-of-line characters.

Business Object Persistence

- New material was added to the topic overview document describing the following topics:
 - *XML Serialization of Data Objects*;

IdeaBlade DevForce Release Notes

- The new *EntityManager.AttachEntity()* method; and
- *Accessing Related Entities Via Navigation Properties on Entities Retrieved Using Stored Procedure Queries*.
- The three existing C# solutions (Console, Silverlight, and WinForm) for *Adding and Deleting* were translated to VB.
- The POCO code solution was translated to VB.

Property Interceptors

- The existing C# solution was translated to VB.

Validation

- A Silverlight application was added, implementing the same verifiers as the previously available WPF and Winform solutions. These include:
 - standard attributed verifiers (generated by the Object Mapper);
 - a DevForce-supplied verifier (DateTimeRangeVerifier);
 - a verifier (BirthDateRangeVerifier) that subclasses a DevForce-supplied verifier (DateTimeRangeVerifier);
 - a verifier (PhoneNumberVerifier) that subclasses a DevForce-supplied RegexVerifier; and
 - a delegate verifier (GetBornBeforeHiredVerifier) that compares the value of two different Employee properties to determine if a rule has been broken.
 - a cross-object verifier that compares the OrderDate on an Order to the HireDate of the Employee listed as the order's SalesRep.
- The WinForm solution for *Validation* was translated to VB.

User Interfaces

- The existing C# WinForm solution, *Composing with User Controls*, was translated to VB.
- A new Silverlight solution, *Grouping and Filtering with the ObjectDataSource*, extends the sample *Using the ObjectDataSource* to add grouping and filtering features.
- A new Silverlight solution, *Editing with the ObjectDataSource*, illustrate editing operations with an ObjectDataSource.

Security

- A new console app, *WindowsAuthentication_ConsoleUI*, was added.
- A new Silverlight sample, *ASP.NET Membership with Registration*, shows how to integrate with the DevForce-provided login features, in addition to using DevForce POCO support to provide new user registration.

Deployment

- A new code snippet, *TraceWindow*, was added showing how to subscribe to the *TracePublisher* and display tracing messages from the Silverlight application.

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Fixed a problem with immediate execution queries having an Include clause in which they did not fetch the related “included” entities. Immediate execution queries are those involving First, FirstOrDefault, Last, LastOrDefault, ElementAt, and ElementOrDefault. [B876]

- The following pre-compiler warning when generating VB developer classes has been fixed:

"warning BC42314: XML comment cannot be applied more than once on a partial class. XML comments for this class will be ignored."

[B1074]

- Fixed a problem in which calling *IEditableObject.CancelEdit* on an Entity did not raise the *PropertyChanged* event(s). [B1075]
- Addressed a problem in which navigation properties on an entity retrieved using by a stored proc query always returned the null entity.

This resulted from a largely undocumented behavior of the Entity Framework's in which it handles entities retrieved via stored procedures differently than those retrieved with normal queries. For the latter, the EF automatically retrieves foreign key values which can later be used to retrieve the related entities. For entities retrieved via stored procedure, the EF does not retrieve the foreign key values.

Recall that in the Entity Framework, *all* related entities must be retrieved by explicit command. When such command is given, EF always returns the related entities, but it uses a different (and less performant) process for parent entities that were retrieved using stored procedures than for entities retrieved using queries against tables.

If you know you will need the related entities for entities retrieved using a stored procedure proc, you can now set a new *ShouldLoadEntityRefs* property on the *StoredProcQuery*. The default for this property is *false* so that the performance hit will not be incurred when you do not, in fact, need to reference the related entities. If you set this property to *true*, all foreign key properties on the entity are looked up during the initial query, and references to related entities will return the proper entities.

[B1078]

- The installation program now automatically installs the latest NorthwindIB, overwriting any prior version [B1082]
- A problem in which the use of fully qualified assembly names in the Web.config could cause config files to grow has been fixed. [B1086]
- Sluggishness in the WPF Trace Viewer was corrected. [B1089]

IdeaBlade DevForce Release Notes

- The exceptions thrown when an EntityQueryFilter is improperly applied to a complex EntityRelationQuery have been made more understandable. [B1091]
- Previously the comment on a Navigation Property indicating whether the property had a getter and/or setter was generated incorrectly. This was fixed. [B1093]
- Previously a NullReferenceException was thrown by EntityProperty.SetValue when no Setter was defined (that is, for readonly properties). A more helpful exception is now raised. [B1105]
- A problem was corrected in which, under certain circumstances, the EntityAspect.EntitySetName property did not get set properly. [B1110]
- Previously, if a client-side OnPropertyChanged handler failed after a save, the EntityManager would be left in an invalid state. The EntityManager will now eat all event errors raised during any load operations (Fetch or Save).

Affected events include *EntityGroup.EntityChanging/Changed*, *EntityGroup.EntityPropertyChanging/Changed*, and *EntityWrapper.PropertyChanged*. [B1111]

- Previously it was not possible to create (and return) null entities for abstract queries. It now is. An abstract query is a query against a polymorphic abstract base class. [B1113]
- A problem was corrected in which related entity fixup did not always work properly with many-many relations. [B1117]
- It is now possible to set the QueryStrategy in the Querying event. [B1119]
- An exception with an explanatory message is now thrown when an entity from one instance of an EntityManager is assigned to a navigation property of an entity from a different EntityManager instance. [B1123]
- A memory leak in the EntityListManager was fixed. The EntityListManager now implements IDisposable and must be Disposed to avoid the leak. [B1124]
- A memoryLeak in the EntityManager.Clear method was fixed. [B1127]
- DevForce-generated simple business object properties use the *GetValue()* and *SetValue()* methods of *DataEntityProperty* to retrieve and set values. (You can see this in the code generated by the Object Mapper.) When *SetValue()* is called, the host entity's *EntityState* gets set to Modified as a byproduct.

It is also possible for you, the developer, to code a custom property so that it, too, uses the services of a *DataEntityProperty*. This is necessary if for any reason you wish to use property interceptors with your custom property.

In one use case brought to our attention, the customer wished to define a custom property using the *DataEntityProperty* approach, and also wished to define a setter for this property. However, exercising the setter caused no persistable changes to the host entity, so he did not want the entity's *EntityState* to be changed because of exercising the setter.

In order to facilitate this use case, we have added an additional constructor for *DataEntityProperty*. This new constructor takes an *isNativeProperty* flag (whose default value is true). The value of *isNativeProperty* is checked prior to setting the *EntityState* of a host entity. If you wish for the setter not to

IdeaBlade DevForce Release Notes

affect the host entity's EntityState, use this constructor for DataEntityProperty (shown below) and pass "false" for the parameter *isNativeProperty*.

```
C# protected DataEntityProperty(Type entityType,
    String propertyName, Type dataType,
    bool isNullable, bool isPartOfKey,
    ConcurrencyStrategy concurrencyStrategy,
    bool isAutoIncrementing, VerificationSetterOptions
    verificationSetterOptions, bool isNativeProperty)
```

[B1129]

- Attributed metadata property verifiers -- for example when defined in a "buddy" class -- previously were not getting discovered. This problem has been fixed. [B1130]
- Previously there was a problem in which dependent foreign keys were not updated when a primary key which was also a foreign key changed. This operation will now throw an exception with an error message that explains that primary keys cannot be updated on an attached entity while relations to that entity exist in the EntityManager. [B1131]
- A problem with queries that return IEnumerable of primitive types within (i.e., as a member of) an anonymous projection was fixed. [B1132]
- Previously if, in the Object Mapper, a developer made the getter access modifier different from the setter access modifier for a collection navigation property, the generated code would contain errors that would cause a build failure. This was fixed. [B1133]
- Previously ImportEntities failed on a detached entity. An exception with explanatory message is now raised. [B1135]
- Previously, updates to properties in a complex type were not saved to the database. This has been fixed. [B1137]
- Fixed a problem which could occur when merging or restoring an EntityCacheState containing temporary Ids. Previously, an ArgumentException with a message stating "Value is not compatible with property" could be thrown in some circumstances, causing the restore to fail. [B1145 – fixed for 5.2.2.1]
- Fixed a problem when an exception stating "You may not change the entity's EntityKey while any of its related, dependent entities are in the entity cache." was erroneously thrown when related entities were loaded into cache. [B1178 – fixed for 5.2.2.1]

WinClient Only

- WinForm control suite assemblies that previously were loaded unnecessarily at run time are now loaded only when needed. This was accomplished by pre-loading binders based on user assemblies. [B1071]

Silverlight-Only

- Previously, if the business model assembly was not referenced in the Silverlight application project, DevForce displayed a warning message about mismatched client and server models. But no reference to the model assembly should be required in a Silverlight application project, and now, none is. [B1096]

- Previously the EntityListManager in Silverlight, throwing security exceptions. We rewrote the WeakEventHandler logic to support the Silverlight security model. [B1104]

Known Issues

Using the Learning Resources under Windows Vista or Windows 7. For those using these operating systems, and depending upon your system security settings, you may need to copy the *Learning Resources* folder installed by DevForce to a new location in order to be able to modify or otherwise work with the code solutions therein. By default, the *Learning Resources* folder is installed under *\Program Files\IdeaBlade DevForce*. You can copy the Learning Resources folder to your personal folder under *Users*, or to any other location to which you have write access.

Installation of NorthwindIB under Windows Vista or Windows 7. On some Vista and Windows 7 machines, during DevForce installation (or stand-alone use of the Database Installer), a dialog appears saying:

Attach of NorthwindIB failed. You may have to manually attach the database.
Click "Ok"; your installation will proceed normally and everything will be fine ...

... except that the *NorthwindIB* tutorial database will not be attached to your SQL Server and you won't be able to run the tutorials and sample code solutions. We are looking into the problem and expect to have it resolved soon.

Note that the database installer will have *detached* an existing instance of this database; it just can't attach the new one.

You can attach the database manually. The following steps are for SQL Server 2005; please adjust for the version of SQL Server you've installed:

- Locate "SQL Server Management Studio" in the Windows start menu.
- Right-click and choose "Run as Administrator" from the context menu. (It is essential that you run as an Administrator if you have Windows UAC turned on.)
- Select the "Databases" node.
- Right-click and choose "Attach ..." from the context menu.
- Press the "Add" button.
- Navigate to "NorthwindIB.mdf" which we will have deposited in the default location, typically "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data".
- Press the "OK" button.

That should do it.

Version 5.2.1

Summary

The primary changes in this release of DevForce are:

- Support for updating POCO objects (Plain Old CLR Objects);
- Support for the parameterization of POCO queries; and
- Updates to address breaking changes between the Silverlight Mix release and the July 9 Silverlight RTM (Release-To-Manufacturing).

Breaking changes

All Products

Server-Side Fetch- and Save-Related Interfaces Moved to *IdeaBlade.EntityModel.Server* Assembly

The *IEntityServerFetching*, *IEntityServerFetched*, *IEntityServerSaving*, and *IEntityServerSaved* interfaces and corresponding event args (*EntityServerFetchingEventArgs*, *EntityServerFetchedEventArgs*, etc) have all been moved into the *IdeaBlade.EntityModel.Server* assembly (and namespace) from the *IdeaBlade.EntityModel* assembly. This may require an additional or alternative “using” (c#) or “Import” (vb) statement to be added to any code that references these interfaces.

Silverlight-Only

The *PropertyName* member of the *PropertyChangedEventArgs* received by a *PropertyChanged* event handler will now be an empty string rather than a null when all properties of the entity have changed. If you are listening for *PropertyChanged* events on the *Entity* (or *EntityWrapper*), you may need to modify your code for this change. This change is related to Defect Repair #B1112 (documented below).

New and Improved

New features are categorized by the DevForce versions to which they apply:

WinClient and Silverlight

- POCO objects are now updatable. [#F1164]
- *EntityQueryParameters* are now supported for POCO queries. [#F1161]
- DevForce no longer throws a FIPS exception when FIPS compliance is turned on. [#F1106]
- During installation, DevForce now automatically uninstalls any prior version (beginning with 5.2.0) that it discovers on the target machine. [#F1159]
- A new *IdeaBlade.Linq.PredicateBuilder* class allows composite predicates to be built up piecewise. [#F1168]

Silverlight-Only

- DevForce was updated to address breaking changes in Silverlight 3 between the MIX release and the RTM version. [#F1156]

- New types were added for use in Silverlight applications: an `ObjectDataSource` control and an `EntityQueryPagedCollectionView`. The `ObjectDataSource` can be used within XAML declaratively to define the data source for bound controls. The `EntityQueryPagedCollectionView` (which is used by the `ObjectDataSource` but which can also be used standalone) provides a collection of entities using asynchronous execution of an `EntityQuery` for paged results. These are both considered beta features and are subject to change. [#F1162]

Learning Resources

Numerous new code solutions have been added; and a number of pre-existing code solutions have been translated to Visual Basic. Additions and changes are listed by category below.

Business Object Mapping

All three existing code solutions associated with the Object Mapping tutorial are now provided in Visual Basic as well as C#.

Business Object Persistence

- *Retrieving Data*. All five existing code solutions in the Retrieving Data tutorial are now provided in Visual Basic as well as C#.
- *Cascading Deletes*. Three solutions illustrate different approaches to implementing cascading deletes in your application.
- *Life Cycle Events*. A new code solution illustrates the use handlers for the client-side events *querying*, *fetching*, *fetched*, *saving*, and *saved*; and for the server-side events *fetching*, *fetched*, *saving*, and *saved*.

Base Applications

This new category was added to provide a place for code samples that are complete (albeit simple) applications that are used as a basis for other code samples and tutorials.

Several application were moved from other categories:

Application	Moved From	Reason
FirstSilverlightApp	User Interfaces\Silverlight	Addresses overall architecture and construction of a Silverlight app, not simply user interface issues.
SilverlightNextSteps	User Interfaces\Silverlight	Addresses overall architecture and construction of a Silverlight app, not simply user interface issues.
LiteHausWPF	User Interfaces\WPF	Addresses overall architecture and construction of a DevForce app. Happens to use a WPF interface; but no UI-specific DevForce facilities are involved, since none exist.

Also, both “FirstSilverlightApp” and “SilverlightNextSteps” are now available in Visual Basic as well as C#.

User Interfaces

- *WinForms*. The app “200WC Configuring a Winform” is now available in Visual Basic as well as C#.
- *Silverlight*. Three new C# applications have been added:
 - “Using a Paged Collection”, and
 - “Using the ObjectDataSource”
 - “Disconnected Out-of-Browser”.

The first two illustrate new features of DevForce which are discussed above in the *New and Improved* Section under the heading “Silverlight-Only”. The “Disconnected Out-of-Browser” sample shows how to do a Silverlight app that runs stand-alone on the client.

- Some applications were moved from this category to Base Applications, as discussed above.

Documentation

Developers Guide

Chapter 3, “Hello DevForce” – The walk-through of the Object Mapper has been reworked to reflect DevForce version 5.2.1 operation and dialog screens.

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- GroupBy no longer throws an erroneous InvalidCastException [#B1076]
- A manifest was added to ServerConsole.exe and ServerService.exe (which launch the DevForce Business Object Server); also to the executable files for the utilities Product Key Updater and Database Installer. All of these programs need to run under Windows with administrator rights: the new manifest causes them to do so *automatically*. [#B1081]
- The operation of EntityWrapper.Equals was changed to handle null entity equality testing correctly. This resolves a problem which occurs when binding to WinForm combo boxes. [#B1082]
- The DevForce installation executable (and the stand-alone Database Installer) now always installs the latest version of NorthwindIB. This ensures that code samples dependent on particular structure or data in the database will run. [#B1083]
- When RemoveEntities() is invoked, the IEnumerable passed into it is now resolved (copied into) into a separate concrete instance (a list) before being processed internally. This avoids problems where an unresolved IEnumerable is passed in and this collection is then modified via deletion. [#B1088]
- The DynamicTypeConverter now respects underlying types for empty collections. [#B1100]
- EntityKeyList now supports abstract entity types. [#B1102]
- A duplicate key exception which could occur during the restore of an EntityCacheState into an EntityManager was fixed. [#B1107]
- DevForce no longer serializes the “proposed values” of an entity when sending to the server. These are needed only for the purpose of canceling an edit locally and so constitute unnecessary baggage in the data payload. [#B1108]

- DevForce now supports serialization of the `System.DateTimeOffset` data type for all applications, including Silverlight apps. This is the auto-generated type in entity and domain models for the new SQL Server 2008 `DateTimeOffset` data type. [#B1109]

Silverlight-Only

- The Object Mapper now correctly sets the `RootNamespace` in Silverlight VB domain model projects to ensure that the domain model and linked Silverlight domain model remain in synch. [#B1106]
- The Object Mapper also adds a reference to the `System.ComponentModel.DataAnnotations` assembly to Silverlight domain model projects for which the “Generate Binding Attributes” flag has been set in the Object Mapper. [#B1106]
- In the `PropertyChangedEventArgs` provided to a `PropertyChanged` event handler, DevForce now uses an empty string instead of null as the `PropertyName` value to indicate that all properties of an entity have changed. This change was made to accommodate some anomalies in the Silverlight 3 `DataForm`. [#B1112]

Appendix. Converting a DevForce Release Candidate Silverlight App To Run Under DevForce 5.2.1 and Silverlight 3 RTW

RC (Release Candidate) versions of DevForce were built around the beta version of Silverlight 3 released at the MIX 09 conference in March 2009. DevForce 5.2.1 is built on the Silverlight 3 RTW version of Silverlight, released July 10, 2009, by Microsoft. That release included breaking changes to which DevForce had to respond.

If you have an existing DevForce Silverlight app built using a Release Candidate version of DevForce, you can update it to DevForce 5.2.1 (and Silverlight 3 RTW) using the following steps, listed here and then detailed in the remainder of this appendix:

1. Fix Up Client-side References in Your Existing App
2. Fix Up Server-side References in Your Existing App
3. Make Sure that `CopyLocal` is set to `True` on all Referenced Server-side IdeaBlade Assemblies
4. Independently, create a new DevForce Silverlight solution (to use for spare parts)
5. In the solution to be converted, replace the `Default.aspx` File
6. In the solution to be converted, add a `Javascript` File to the Web Project
7. Edit the `Object` tag in `Default.aspx` to specify the correct XAP file
8. Compile and replace any Silverlight statements that were obsoleted by the release to their release-version equivalents
9. Test the updated solution

Step 1. Fix Up Client-side References in Your Existing App

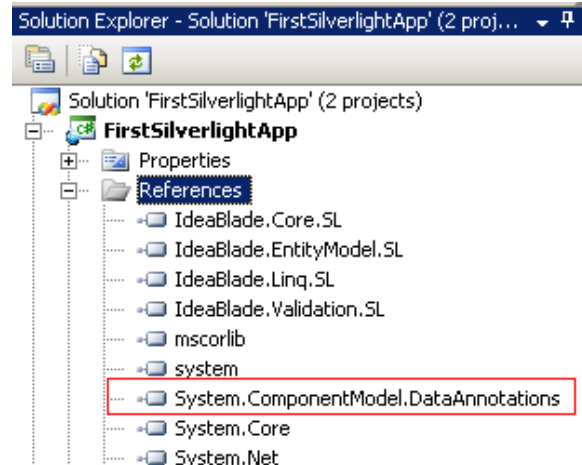
In the client-side project of the existing solution you wish to update, remove any references to the following assemblies:

IdeaBlade DevForce Release Notes

- System.ComponentModel and
- System.ComponentModel.DataAnnotations

If System.ComponentModel.DataAnnotations was previously referenced, re-add it to get the newer version:

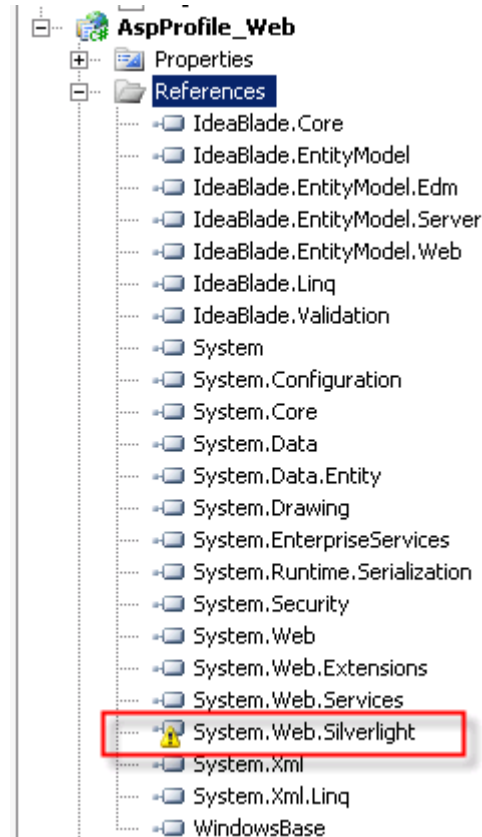
- System.ComponentModel.DataAnnotations



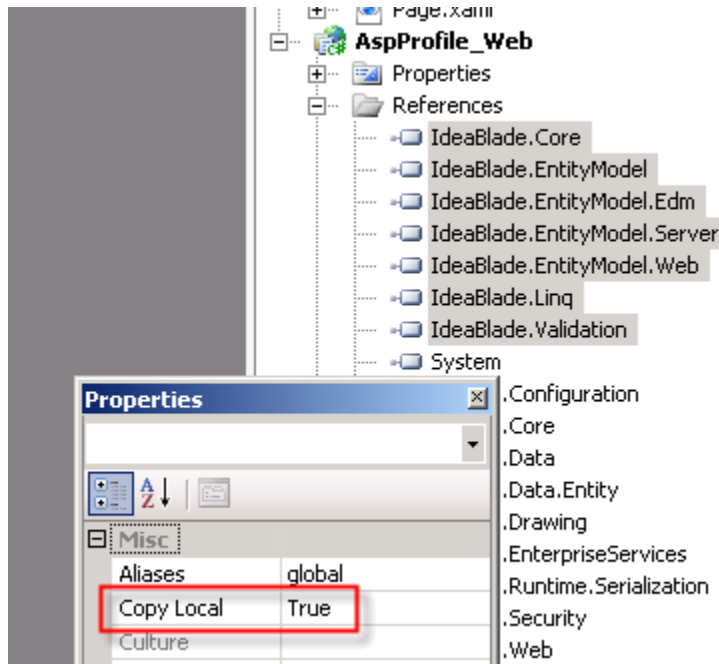
Step 2. Fix Up Server-side References in Your Existing App

In the server-side project of the existing solution you wish to update, remove any reference to the following assembly:

- System.Web.Silverlight



Step 3 . Make Sure that CopyLocal is set to True on all Referenced Server-side IdeaBlade Assemblies

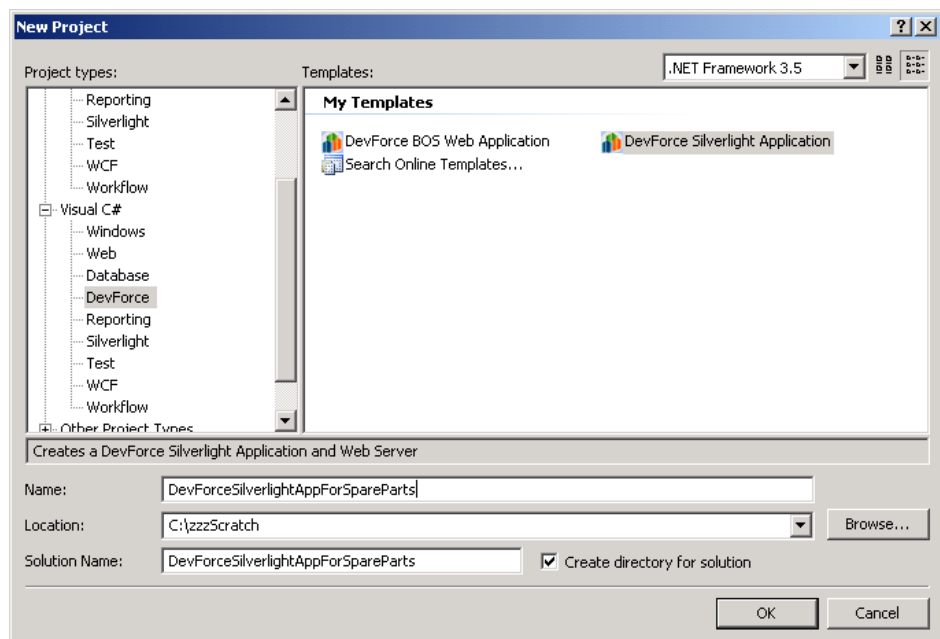


Step 4. Independently, create a new DevForce Silverlight solution (to use for spare parts)

In Visual Studio, create a new solution using the DevForce Silverlight Application project template.

This will generate some files we'll cannabalize to update our existing DevForce Silverlight app.

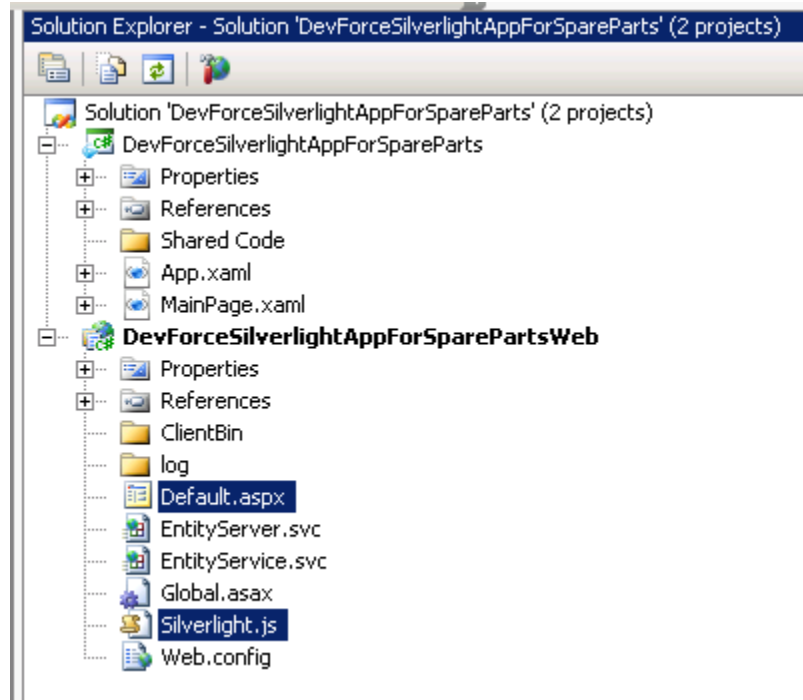
For added convenience, name this solution the same as your to-be-converted existing solution.



IdeaBlade DevForce Release Notes

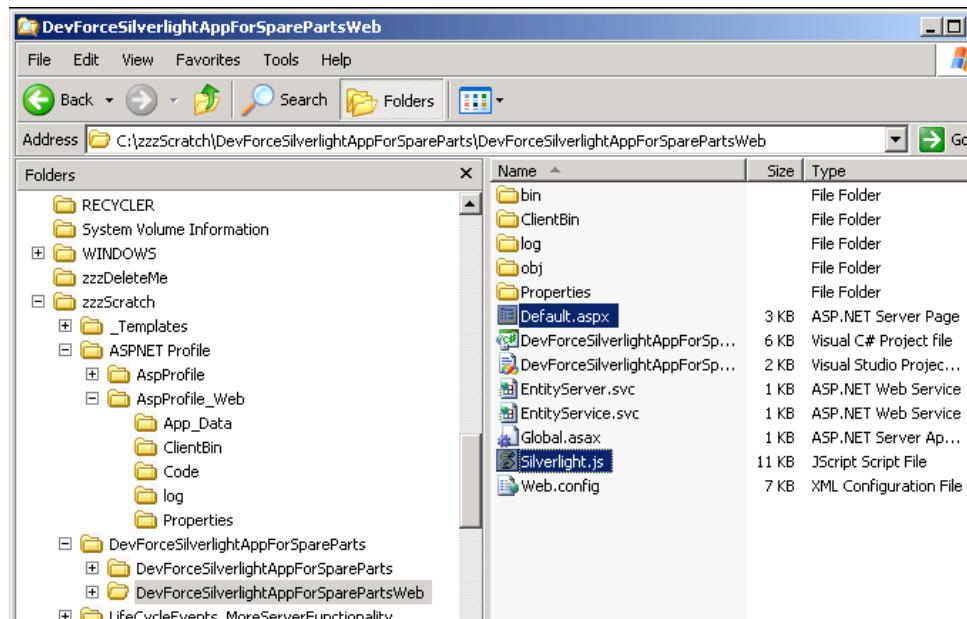
The template creates two projects: one a web project, to be deployed server-side, and one a Silverlight-project, to be deployed client-side.

The files we'll need are both in the web (server-side) project: they are **Default.aspx** and **Silverlight.js**.

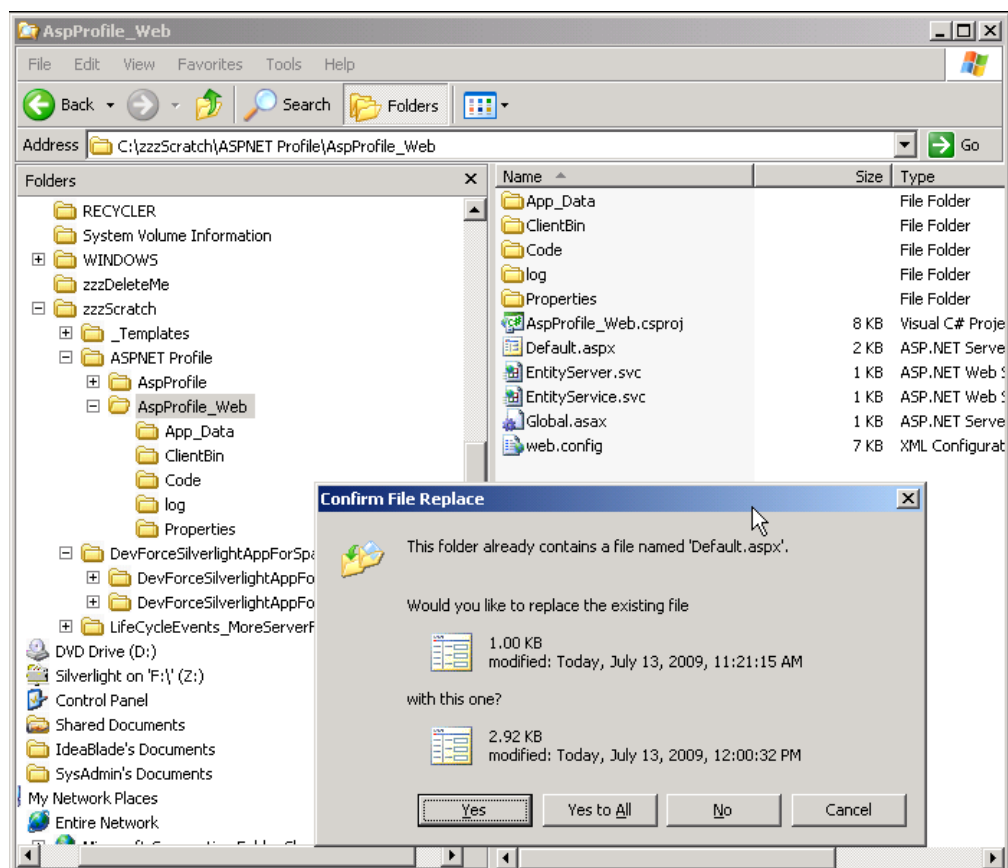


Step 5 . Cannabalize Default.aspx and Silverlight.js

Copy Default.aspx and Silverlight.js from the web project folder in the “spare parts” solution to the corresponding folder in the solution to be converted. Note that you’ll be overwriting an existing copy of Default.aspx.

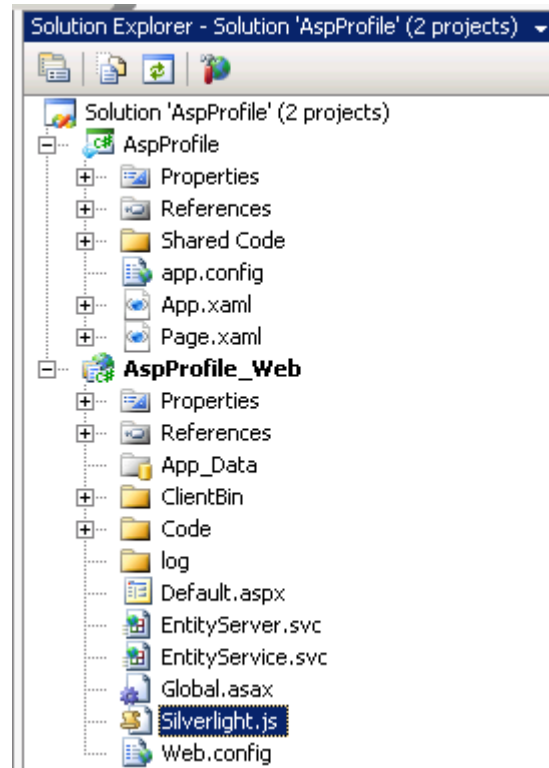


IdeaBlade DevForce Release Notes



Step 6. In the solution to be converted, add a Javascript File to the Web Project

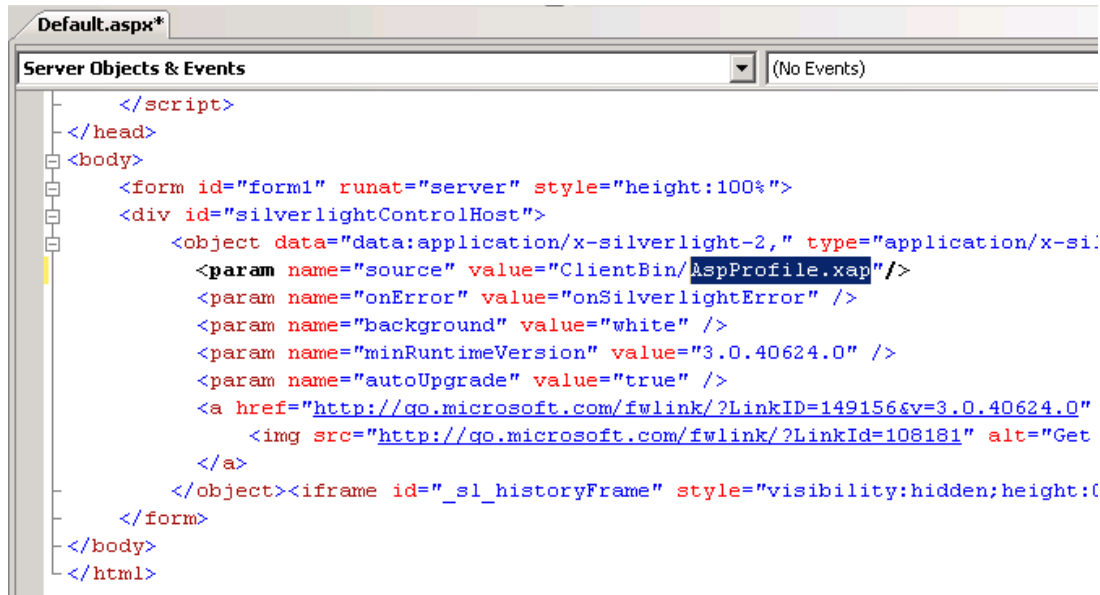
Add Silverlight.js to the web project of the existing solution, using Add / Existing Item.



Step 7. Modify Default.Aspx

Edit the Object tag in Default.Aspx to specify the correct XAP file (see below). Save your change.

Note: if you named your solution-to-be-cannabalized the same as your existing solution, you won't need to edit the XAP file name in the Object tag: it will already be correct.



Step 8. Compile and replace any Silverlight statements that were obsoleted by the Silverlight RTW release to their RTW equivalents

For this step, consult appropriate Silverlight change docs or release notes from Microsoft and others. See, for example, the Silverlight Toolkit July 2009 changes at

<http://silverlight.codeplex.com/Wiki/View.aspx?title=Silverlight%20Toolkit%20July%202009%20change%20list>

and the Microsoft document, “Ensuring That Your Silverlight 2 Applications Work with Silverlight 3” at

[http://msdn.microsoft.com/en-us/library/cc645049\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/cc645049(VS.95).aspx)

Step 9. Test the updated solution

1. Open the DevForce Object Mapper and regenerate the domain model.
2. Set the web project to be the solution’s Startup Project.
3. Run the solution

Version 5.1.1

Summary

The primary changes in this release of DevForce are new support for Plain Old CLR Objects (POCOs) and integration with ASP.NET Security. Please see the New and Improved section below for a full list of enhancements.

Breaking changes

There are a number of breaking changes in this release. Most are necessary consequences of implementing the new support for Plain Old CLR Objects (POCOs) or integrating with ASP.NET Security. (See New Features for details on these.)

Extension Methods Returned EntityQuery<T> Retired

IdeaBlade DevForce Release Notes

Specific extension methods targeted at `EntityQuery<T>` as opposed to `IEntityQuery<T>` have been removed in favor of their `IEntityQuery<T>` counterparts. This was done both in the interest of simplicity and to follow the pattern adopted by the standard LINQ extensions to `IQueryable<T>`.

The effect of this change is that any extension operation on an `EntityQuery<T>` will now return an `IEntityQuery<T>`.¹ This means that any code written in the following form...

```
C# EntityQuery<Customer> customers = myEntityManager.Customers.Where(...)
```

...will need to be replaced by either...

```
C# IEntityQuery<Customer> customers = myEntityManager.Customers.Where(...)
```

...or...

```
C# var customers = myEntityManager.Customers.Where(...)
```

EntityManager and Entity Methods Now Return “Object” Type, Not “Entity”

Any `EntityManager` or `Entity` methods that formerly had a return type of ‘Entity’ now have a return type of ‘Object’. This change was required to permit the `EntityManager` to manage POCO objects (that may or may not inherit from the `IdeaBlade.EntityModel.Entity` class).

Note that the result returned has not changed: only the declared return type has changed. The methods affected are the non-generic overloads of the following methods:

- `EntityManager.CreateEntity`
- `EntityManager.FindEntity`
- `EntityManager.FirstOrNullEntity`
- `EntityManager.GetNullEntity`

The generic versions of these methods have not changed.

Because of the change, statements such as the following...

```
C# Entity myEntity = EntityManager.FindEntity(...);
```

...will need to be modified to look like this:

¹ Similarly, any extension operation on a concrete implementation of `IQueryable<T>` will always return an `IQueryable<T>`.

```
C# Entity myEntity = (Entity) EntityManager.FindEntity(...);
```

EventArgs With Entity Properties Will Now Have Object Properties

Any EventArgs that contain a property with a return type of Entity will now have a return type of Object. Because of this change, code of the following form...

```
C# var handler = new EventHandler<EntityFetchedEventArgs>((o, e) => {  
    bool isModified = e.EntityState.IsModified();  
})
```

...will need to be replaced with...

```
C# var handler = new EventHandler<EntityFetchedEventArgs>((o, e) => {  
    bool isModified = ((Entity) e).EntityState.IsModified();  
})
```

...or...

```
C# var handler = new EventHandler<EntityFetchedEventArgs>((o, e) => {  
    bool isModified = EntityWrapper.Wrap(e).EntityState.IsModified();  
})
```

EntityQuery 'ReturnType' property Has Been Retired

The EntityQuery 'ReturnType' property was a duplicate of 'ElementType' property and has been removed.

Logout Method Added to the *IEntityLoginManager* Interface

The IEntityLoginManager interface now includes a Logout() method. Implementers of this interface will need to add a method for the following signature:

```
C# void Logout(IPrincipal principal, EntityManager entityManager);
```

If not Logout operations are required, simply implement an empty method.

New and Improved

New features are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

POCO Support (Beta Feature)

DevForce now supports the use of Plain Old CLR Objects (POCOs), above and beyond its support for the Entity-based objects generated by its Object Mapper. The POCO support is documented in the Object Persistence chapter of the Developers Guide, in the section "POCO Support in DevForce".

EntityManager Has a New ‘Querying’ Event

```
C# public event EventHandler<EntityQueryingEventArgs> Querying;
```

This event was added to address the requirement to modify a query prior to sending it to the server. The EntityManager Fetching event does not allow this because the Query argument provided to a fetch can be a subset of initial query submitted because of client-side query optimization; and modification of an optimized query is therefore not allowed. The Querying event, by contrast, contains a modifiable (and cancellable) version of the actual query submitted. [#F1067]

New Configuration Flag to Determine Which Serializer is Used

A new attribute has been added to the ObjectServer element in the ideablade.configuration section called *UseDCS*. This flag controls whether the DataContractSerializer (*useDCS="true"*) or the NetDataContractSerializer (*useDCS="false"*) will be used in communications with the DevForce Business Object Server. By default, Silverlight applications will always use DCS and this flag will be ignored. In other applications, the default is to use the NetDataContractSerializer, since it is more lenient with serializing type information. Note that use of the NetDataContractSerializer also results in a somewhat larger message size.

You will find further discussion of the serializers in the Object Persistence chapter of the Developers Guide, in the section “POCO Support in DevForce”. [#F1124]

DevForce Now Integrates with ASP.NET Security Services (Membership, Role, and Profile).

See the Developer’s Guide, Security chapter, section “ASP.NET Security Integration”, for more information. [#F1137]

Logout Method Added to the IEntityManager interface.

Any custom processing needed when a user session terminates can be included here. [#F1141]

New Authorization Attributes Have Been Added.

New declarative security attributes - *RequiresAuthentication* and *RequiresRoles* – can be used to adorn server-side methods to perform user authorization prior to executing the method. See the Developer’s Guide, *Security* chapter, section “Authorization Attributes”, for more information. [#F1142]

Support Added for a Light-Weight Login Mechanism for ASP.NET solutions. (Beta Feature)

A new EntityManager.Login overload was added that accepts a session key:

```
C# public bool Login(Guid sessionKey);
```

A session key is only available after a successful login via the new EntityManager *SessionKey* property.

This mechanism was added to allow ASP.NET server applications to store the session key after a successful login and reuse it for future logins without having to resubmit full credentials every time. Note that this feature will NOT work in a load balanced environment, whereas the standard Login with credentials will. [#F1145]

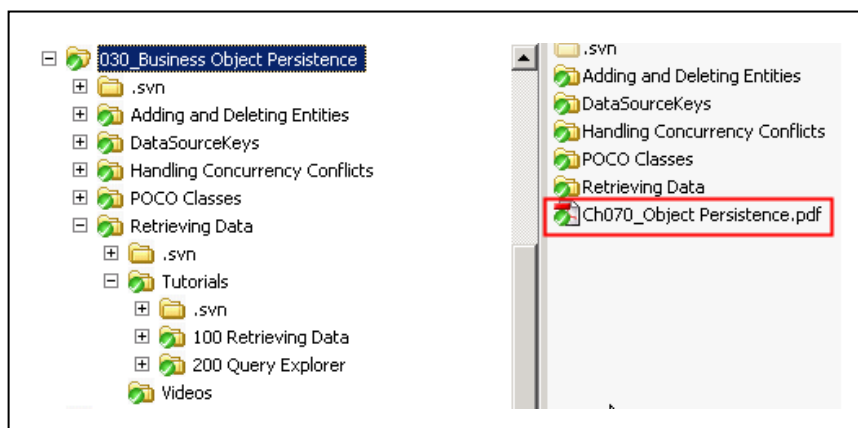
WinClient-Only Features:

New Third-Party Control Suite Versions Supported

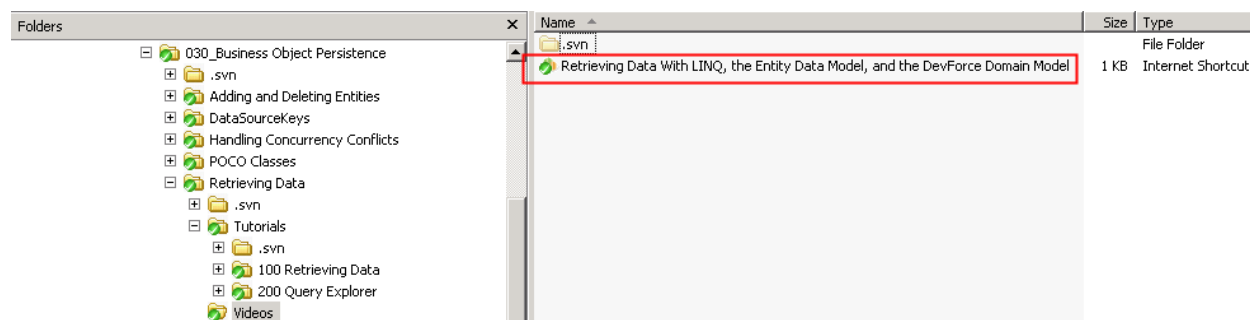
DXperience v2009 vol.1 and *Infragistics NetAdvantage v2009 vol.1* WinForm control suites are now supported by DevForce.

Learning Resources (formerly “Learning Units”)

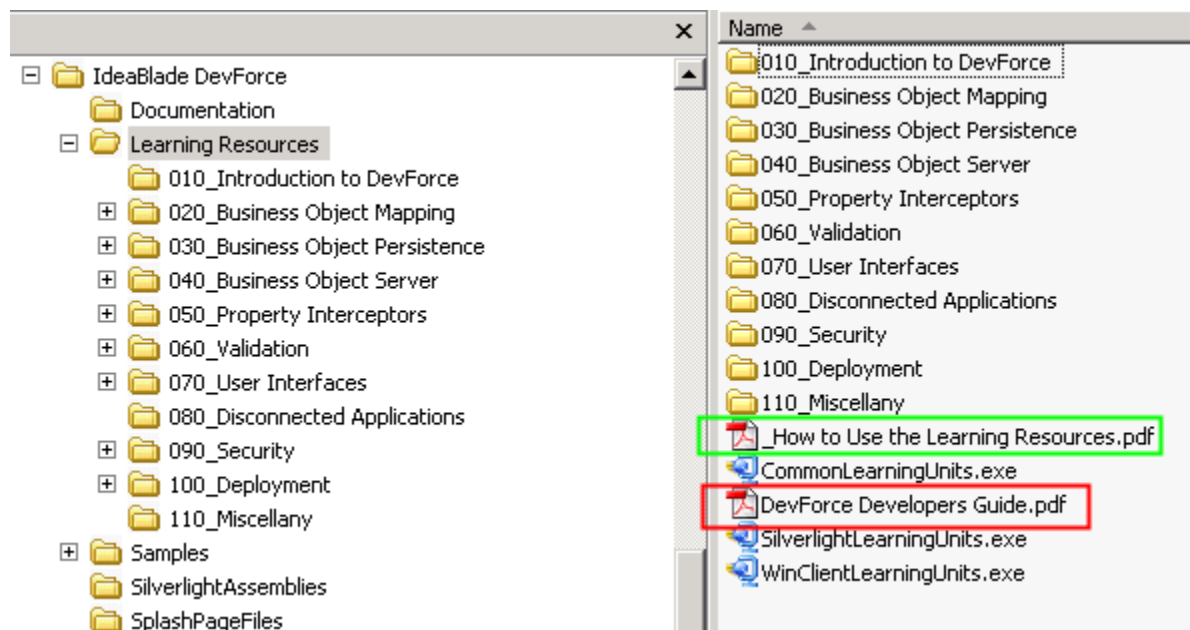
The Learning Units have been reorganized by Topic → SubTopic → Level → Technology (e.g., WinClient, Silverlight, or Both); and additional resources have been included within topics, including videos and the appropriate chapter from the Developers Guide. Below, for example, see the Business Object Persistence folder. In the the root is the Object Persistence chapter from the Developers Guide, produced as a stand-alone Adobe Acrobat file. Below the root you see subtopic folders (“Adding and Deleting Entities”, “DataSourceKeys”, etc.); within each of these you will find documented code solutions and tutorials.



Links are also provided to videos deployed on the IdeaBlade web site and elsewhere. The links are placed under the topic or subtopic that they most directly target.



In the root of the Learning Resources folder you will find a single-page version of the Developers Guide that contains links to the individual chapters, in the same order that they appear in the complete, single-document Developers Guide. This is the document highlighted in red below:



Note also the document “How to Use the Learning Resources.pdf” (highlighted in green) which contains useful tips for making best use of the resources.

Documentation

Developers Guide

The Developers Guide is now available in two forms:

- As a complete, single document, linked from the Windows Start Menu for IdeaBlade DevForce; and
- As individual chapters, placed under their appropriate topic in the Learning Resources.

Accompanying the latter is a single-page document, deployed in the root of the Learning Resources folder, that links to the individual chapters, in the same order that they appear in the complete Developers Guide.

Chapter 7, “Object Persistence” - DevForce’s new support for Plain Old CLR Objects (POCOs) is documented in the section “POCO Support in DevForce”.

Chapter 14, “Security” - DevForce’s new integration with ASP.NET security facilities is now documented; implementation of the `IEntityLoginManager` interface is discussed; and new material is presented on using server-side persistence events.

Four Simple Steps Article and Solution

The tutorial-like character of this article (first introduced last release) has been extended into the User Interface section to make it easier for a reader to retrace the complete steps to produce the solution. The article and solution have been moved into the Learning Resources folder (where other code solutions and tutorials reside); but are still linked from the splash page that displays at the end of DevForce installation. (That page, in turn, is also linked from the Windows Start Menu for IdeaBlade DevForce.)

Defect Repairs

Defect repairs are categorized by the DevForce versions to which they apply:

WinClient/Silverlight

- Change EntityWrapper.Equals to better handle Null Entity. [#B1082]
- Support ConditionalExpression on EntityQuery. Queries involving conditional operators now serialize properly. Queries similar to the one below now operate correctly. [#B1054]

```
C# var x = from e in _EntityManager.TrainingCourses
        select new { Disabled = (e.Accredited.Value ? 1 : 0) };
```

- DevForce now does not install Silverlight project templates if the installation license key used is for WinClient [#B1036]
- Install no longer fails when Visual Studio has never been initialized [#B1051]
- WinClient Install will now install Classic assemblies even if Classic is already installed [#B1059]
- You may now install DevForce to a non-empty folder [#B1070]
- The SaveOption.FixupTempIds default used when saving a list of entities with store-generated IDs is now InSaveListOnly. This change affects only the following overloads:

```
C# public SaveResult SaveChanges(IEnumerable entities);
    public void SaveChangesAsync(IEnumerable entities,
```

When store-generated IDs are used, the database provides the real ID at the time the item is saved. Because of this, a default option to perform fixup on all IDs (FixupTempIds.All) when only a subset is saved does not make sense, and causes an exception to be thrown if used. When real IDs are supplied by an IIdGenerator, fixup on all temporary IDs, even though a subset of entities will be saved, is usually possible and the default FixupTempIds.All can be used. For this reason, the default option was changed when the entity list contains items with store-generated IDs. [#B1026]

- Throw an exception with an explanatory message when an FK-to-PK column mismatch is found during save processing. [#B1055]
- Custom LoginExceptions thrown from IEntityLoginManager.Login() can now be caught by the client. Previously these exceptions were always coerced into a LoginException. [#B1061]
- The LicenseException message now contains additional explanatory text indicating how to diagnose the problem cause. [#B1062]
- Allow an IBEDMX file to be saved in a project sub-folder. A new file will be saved to the project directory, but if moved to a sub-folder the Object Mapper will save the file correctly. (Note that if you move the IBEDMX file you should manually edit the XML contents to change the relative path to the EDMX file.) [#B1064]
- The properties of the EntityServerException prefixed with “Remote”, such as RemoteExceptionDetails, are now correctly serialized and are available to the client application. [#B1069]
- Object Mapper now supports SQL Server Integration Services projects (projects of extension “.dtproj”). [#B1063]

IdeaBlade DevForce Release Notes

- The Object Mapper would not launch if a project of type “.dtproj” (SQL Server Integration Services project) existed in the solution. This is now fixed. [#B1063]

Silverlight-Only

- The app.manifest generated with the DevForce Silverlight Application template now includes the same out-of-browser information as does the standard Silverlight template. [#B1041]

Version 5.1.0

Summary

The DevForce WinClient and DevForce Silverlight products are now available in a single package called DevForce Universal. A separate license key will need to be purchased to enable the additional functionality.

Other features in this release include:

- Usability enhancements
- Additional learning units and updated documentation
- Improved network performance for Silverlight by using binary serialization.
- Query filters have been introduced that can help refactor code and/or enforce security on the server.

Please see the New and Improved section below for a full list of enhancements.

Breaking changes

There are an unusually large number of breaking changes in this release due to the beta status of DevForce Silverlight and to the rollout of a new generation product.

“WinClient/Silverlight” breaking changes address functionality common to *DevForce WinClient* and *DevForce Silverlight*. “Silverlight-Only” breaking changes address functionality only relevant to users of *DevForce Silverlight*.

WinClient / Silverlight

Namespace and Assembly Name Changes

DevForce assemblies were renamed for this release. In brief, the changes were as follows:

- the .v4 suffix (e.g., *IdeaBlade.EntityModel.v4*) was dropped from all namespaces, assemblies, and executables
- *IdeaBlade.Util.v4* became *IdeaBlade.Core*
- *IdeaBlade.Verification.v4* became *IdeaBlade.Validation*; and
- the old Winforms-based TraceViewer became the *WinTraceViewer* (to distinguish it from the newer WPF version of the same).

The following two tables provide a comprehensive list of the changes:

Old Namespace Name	New Namespace Name
IdeaBlade.Asp.v4	IdeaBlade.Asp
IdeaBlade.DevTools.TraceViewer.v4	IdeaBlade.DevTools.TraceViewer

IdeaBlade DevForce Release Notes

IdeaBlade.DevTools.WPFTTraceViewer.v4	IdeaBlade.DevTools.WPFTTraceViewer
IdeaBlade.EntityModel.v4	IdeaBlade.EntityModel
IdeaBlade.EntityModel.SL	IdeaBlade.EntityModel.SL
IdeaBlade.EntityModel.Edm.v4	IdeaBlade.EntityModel.Edm
IdeaBlade.EntityModel.Server.v4	IdeaBlade.EntityModel.Server
IdeaBlade.EntityModel.WS.v4	IdeaBlade.EntityModel.WS
IdeaBlade.Linq.v4	IdeaBlade.Linq
IdeaBlade.Linq.SL	IdeaBlade.Linq.SL
IdeaBlade.Rdb.v4	IdeaBlade.Ado
IdeaBlade.Util.v4	IdeaBlade.Core
IdeaBlade.Util.SL	IdeaBlade.Core.SL
IdeaBlade.Verification.v4	IdeaBlade.Validation
IdeaBlade.Verification.SL	IdeaBlade.Validation.SL

Old Assembly Name	New Assembly Name
IdeaBlade.Asp.v4	IdeaBlade.Asp
IdeaBlade.EntityModel.v4	IdeaBlade.EntityModel
IdeaBlade.EntityModel.SL	IdeaBlade.EntityModel.SL
IdeaBlade.EntityModel.Edm.v4	IdeaBlade.EntityModel.Edm
IdeaBlade.EntityModel.Server.v4	IdeaBlade.EntityModel.Server
IdeaBlade.EntityModel.WS.v4	IdeaBlade.EntityModel.WS
IdeaBlade.Linq.v4	IdeaBlade.Linq
IdeaBlade.Linq.SL	IdeaBlade.Linq.SL
IdeaBlade.Rdb.v4	IdeaBlade.Ado
IdeaBlade.Util.v4	IdeaBlade.Core
IdeaBlade.Util.SL	IdeaBlade.Core.SL
IdeaBlade.Verification.v4	IdeaBlade.Validation
IdeaBlade.Verification.SL	IdeaBlade.Validation.SL
ConfigEditor.exe	ConfigEditor.exe
DeploymentTester.v4.exe	DeploymentTester.exe
ProductKeyUpdater.exe	ProductKeyUpdater.exe
ServerConsole.v4.exe	ServerConsole.exe
ServerService.v4.exe	ServerService.exe
TraceViewer.v4.exe	WinTraceViewer.exe
WPFTTraceViewer.v4.exe	WPFTTraceViewer.exe

Regenerating your Object Model will add the minimal set of needed new assembly references (consisting of IdeaBlade.Core, IdeaBlade.EntityModel, and IdeaBlade.Validation) to your DomainModel project; you can then

IdeaBlade DevForce Release Notes

delete the corresponding old references (to IdeaBlade.Util.v4, IdeaBlade.EntityModel.v4, and IdeaBlade.Verification.v4) from that project. In your executable project you will need to add the new references manually, then again delete the old ones. Repeat for any other projects in your solution that reference DevForce assemblies.

Namespaces Now Explicitly Declared in Generated Code (Visual Basic Only)

A Namespace statement will now be generated in VB code *if the Domain Model namespace, as specified in the Object Mapper, differs from the root namespace of the assembly*. This change allows you to place the Domain Model and Entity Model in the same assembly, which is a common use case in simpler Silverlight applications. *Imports* statements in your existing code that target the Domain Model should be changed to reflect this practice. [#B1020]

App.Config Changes

The <IdeaBlade.v4> section in the app.config (and web.config) was renamed to <ideablade.configuration>.

This is a breaking change; however, regenerating your business model (to update the app.config in the DomainModel project) and then building your app (to copy changes from the app.config in the DomainModel project to the app.config in the executable project) should be enough to bring your app up to the new standard.

The above process will leave the old <ideablade.v4> section in the app.config file, so that any changes you introduced manually into the section will be preserved. You will need to move such manual changes into the new <ideablade.configuration> section.

You can then delete the old sections, shown in bold and maroon in the following XML snippet:

XML

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <section name="ideaBlade.v4" type="IdeaBlade.Util.Configuration.v4.IdeaBladeSection,
IdeaBlade.Util.v4, Version=4.2.3.1, Culture=neutral, PublicKeyToken=287b5094865421c0" />
    <section name="ideablade.configuration" type="IdeaBlade.Core.Configuration.IdeaBladeSection,
IdeaBlade.Core, Version=5.1.0.0, Culture=neutral, PublicKeyToken=287b5094865421c0" />
  </configSections>
  <ideaBlade.v4 version="4.00" updateFromDomainModelConfig="Ask">
    <logging logFile="DebugLog.xml" />
    <objectServer isDistributed="false" remoteBaseUrl="http://localhost" serverPort="9009"
serviceName="EntityService" />
    <edmKeys>
      <edmKey name="Default"
connection="metadata=res://ServerModelNorthwindIB/ServerModelNorthwindIB.csdl|res://ServerM
odelNorthwindIB/ServerModelNorthwindIB.ssdl|res://ServerModelNorthwindIB/ServerModelNorthw
indIB.msl;provider=System.Data.SqlClient;provider connection string=&quot;Data Source=.;Initial
Catalog=NorthwindIB;Integrated Security=True;MultipleActiveResultSets=True&quot;;"
containerName="ServerModelNorthwindIB.ServerModelNorthwindIBContext">
        <probeAssemblyNames>
          <probeAssemblyName name="DomainModel" />
          <probeAssemblyName name="ServerModelNorthwindIB" />
        </probeAssemblyNames>
      </edmKey>
    </edmKeys>
  </ideaBlade.v4>
```

```
<ideablade.configuration version="5.00" updateFromDomainModelConfig="Ask">
  <logging logFile="DebugLog.xml" />
  <objectServer isDistributed="false" remoteBaseUrl="http://localhost" serverPort="9009"
serviceName="EntityService" />
  <edmKeys>
    <edmKey name="Default"
connection="metadata=res://ServerModelNorthwindIB/ServerModelNorthwindIB.csdl|res://ServerMod
elNorthwindIB/ServerModelNorthwindIB.ssd|res://ServerModelNorthwindIB/ServerModelNorthwindIB.
msl;provider=System.Data.SqlClient;provider connection string='Data Source=.;Initial
Catalog=NorthwindIB;Integrated Security=True;MultipleActiveResultSets=True'""
containerName="ServerModelNorthwindIB.ServerModelNorthwindIBContext">
    <probeAssemblyNames>
      <probeAssemblyName name="DomainModel" />
      <probeAssemblyName name="ServerModelNorthwindIB" />
    </probeAssemblyNames>
  </edmKey>
</edmKeys>
</ideablade.configuration>
</configuration>
```

Silverlight-Only

All communications between a Silverlight client and the BOS are now encoded in binary format by default. This improves network transport speed substantially.

For your existing Silverlight applications, this is a breaking change which must be addressed in the web.config file. To bring your web.config file up to the current standard, replace the existing the <system.serviceModel> node in the web.config file with the following snippet. A sample we.config is also available in the *Samples\Deployment\Silverlight* folder in the DevForce installation folder (typically *C:\Program Files\IdeaBlade DevForce*).

XML	
	<pre><system.serviceModel> <services> <service name="EntityService"> <endpoint address="" binding="customBinding" bindingConfiguration="customBinaryBinding" contract="IdeaBlade.EntityModel.IEntityServiceContract" /> </service> <service name="IdeaBlade.EntityModel.Server.EntityServer"> <endpoint address="" binding="customBinding" bindingConfiguration="customBinaryBinding" contract="IdeaBlade.EntityModel.IEntityServerContract" /> </service> </services> <bindings> <!--The default binding in the Silverlight app is http with binary encoding. Max values are set here for testing; you should adjust to your application's requirements. If you modify the binding here, be sure to also create/modify the ServiceReferences.ClientConfig in the Silverlight application too. Both the client and server sides must agree on the binding configuration in order for communications to succeed. --> <customBinding></pre>

```
<binding name="customBinaryBinding">
  <binaryMessageEncoding>
    <readerQuotas maxDepth="2147483647" maxStringContentLength="2147483647"
      maxArrayLength="2147483647" />
  </binaryMessageEncoding>
  <httpTransport maxReceivedMessageSize="2147483647" maxBufferSize="2147483647" />
</binding>
</customBinding>

</bindings>
</system.serviceModel>
```

New and Improved

“WinClient/Silverlight” features address functionality common to *DevForce WinClient* and *DevForce Silverlight*. “Silverlight-Only” features address functionality only relevant to users of *DevForce Silverlight*.

WinClient/Silverlight

- An *EntityQueryFilter* class was added to DevForce. You now have the ability to inject filters into an *EntityQuery* on either client or server side. For more information see the section “Filtering Queries” in Chapter 7 of the Developers Guide, “Object Persistence”. [#F937]
- Double-clicking the .ibedmx file will now launch the Object Mapper. [#F1112]
- We have made it more obvious when the Object Mapper’s *save* operation completes. [#F1024]
- A new *EntityManager.Principal* property has been added. This property returns the *IPrincipal* representing the authenticated user whose credentials were used to log into the *EntityManager*. [#F1089]
- The generation of Verification attributes can now be controlled at the property level from within the Object Mapper. [#F1090]
- An *IsForeignKey* property is now available on all *DataEntityProperty* subclasses. [#F1123]
- The *EntityKey* type is now serializable. (via WCF). [#F1061]
- Summary tags are now autogenerated into the domain model classes based on documentation annotations in the EDMX file. [#F963]

WinClient-Only Features:

- The Developer Express DXperience v2009 vol.1 control suite is now supported.
- Infragistics NetAdvanced 2009 vol.1 control suite is now supported.

Silverlight-Only Features:

- The new Silverlight 3 binding attributes are generated for Silverlight apps, by default. [##F1069]
- Fault exception handling has been improved. [#F1078]
- A WCF *ChannelFactory* is now used instead of generated proxy classes. [#F1104]

- A ServiceReferences.ClientConfig file is now supported to allow for customization of communications to the BOS. See the example provided in the Samples\Deployment\Silverlight folder of your installation. [#F1105]

Learning Units

For this release the Learning Units have been segregated into Common, WinClient-Only, and Silverlight-Only branches. Learning Units in the Common branch apply to users both of DevForce WinClient and DevForce Silverlight.

Learning Units in the WinClient-Only branch are installed when you install with a DevForce WinClient or DevForce Universal license key, but not when you install with a DevForce Silverlight license.

Learning Units in the Silverlight-Only branch are installed when you install with a DevForce Silverlight or DevForce Universal license key, but not when you install with a DevForce WinClient license.

New or improved Learning Units are noted below.

Common

Adding Deleting (200 Intermediate). This unit illustrates the basics of implementing the capabilities to add and delete entities in your DevForce applications. It uses a simple Console app interface. The same functionality illustrated herein is also addressed in the Adding Deleting unit in the Silverlight-Only group (described below); that unit simply wires the functionality up to a Silverlight user interface.

Security_ConsoleUI (300 Advanced). This unit illustrates the basics of using DevForce authentication facilities; adding security role information to the authenticated IPincipal returned from the login operation; and querying those security roles in the client application. It uses a Console app interface.

WinClient-Only

Security_ConsoleUI (300 Advanced). This unit illustrates the basics of using DevForce authentication facilities; adding security role information to the authenticated IPincipal returned from the login operation; and querying those security roles in the client application. It uses a Console app interface.

DataSourceKeys. This unit illustrates the use of DevForce DataSourceKeys for selecting specific instances of a particular database at runtime (as when multiple “tenants” use your application and have their own copies of a particular database). One solution illustrates the technique for specifying the connection information statically in the app.config file; another shows how to look up the connection information in a database at runtime.

Silverlight-Only

Two new Learning Units are added in this release for DevForce Silverlight users. You will find both in the 200 Intermediate folder.

Model Explorer (200 Intermediate). This Learning Unit updated (to run under this 5.1.0 release) the Model Explorer solution demonstrated at the March 2009 MIX conference and made available for download on the IdeaBlade web site at

http://www.ideablade.com/DevforceSilverlight/DevForceSilverlight_ModelExplorer.aspx

The version of the Model Explorer displayed on the web site may continue to evolve in the direction of greater sophistication, whereas this Learning Unit version will likely remain as is. A brief tutorial script is included to guide you to important parts of the solution.

Adding Deleting (200 Intermediate). This unit builds upon the Silverlight solution included in the 100-series Learning Unit, “First Silverlight Application”, providing Add, Delete, and form-view Editing capabilities to that

solution. The non-UI functionality illustrated herein is the same as in the Adding Deleting unit in the Common group (described above).

Documentation

Developers Guide

Entire guide swept to replace references to old .v4 assemblies with current names.

Chapter 7, “Object Persistence”. A new section, “Filtering Queries”, was added to to address the use of the new EntityQueryFilter class.

Chapter 9, “DevForce Silverlight Apps”. Material added on deployment and troubleshooting.

Four Simple Steps Article in Program Files\IdeaBladeDevForce\Samples

Updated to include Visual Basic code. Visual Basic solution added.

Defect Repairs

“WinClient/Silverlight” defects address functionality common to *DevForce WinClient* and *DevForce Silverlight*. “Silverlight-Only” defects address functionality only relevant to users of *DevForce Silverlight*.

WinClient/Silverlight

- Fixed bug where FindEntityGraph does not find deleted children [##B915]
- Fixed issue where a projection into a custom type failed. [##B1028]
- Fixed issue where a second RejectChanges failed with a duplicate key error when the FK was part of the PK. [##B1021]
- A mismatch between the Object Manager project selection and code generation was corrected. [##B990]
- Fixed issue where collection projection results selected from a relation property did not always serialize properly. [##B957]
- Fixed issue with identity columns not being recognized where inheritance was involved. [##B1001]
- Fixed issue with an ID generation exception sometimes occurring after an ImportEntities call. [##B947]
- Fixed bug where the IdeaBlade AppUpdater wiped out the system.web section of the web.config if no IdeaBlade section was present. [##B991]
- Fixed bug where creating a new domain model project would not update the selected project combo box. [##B982]
- Fixed bug where multiple accesses to the same PendingEntityList would cause multiple fetches. [##B984]
- Fixed bug with nag screen not disappearing after purchase. [##B987]
- Fixed bug where refetch could fail when inherited types were involved. [##B949]
- Fixed bug where a query with multiple roots would sometimes fail to auto-invert. [##B956]
- Custom LoginCredential or ILoginCredential instances will now be correctly reconstructed when IIS recycles. [##B930]
- The Exit menu item now works in the WPFTraceViewer utility. [##B935]
- Renaming of entity type names in the Object Mapper should now work as expected. Any files linked in Silverlight projects will also continue to work. [##B951]
- The SQL Server 2008 Time data type is now supported. [##B985]
- A NullReferenceException will not be thrown when the NoLoginManagerFound LoginException is expected. [##B992]
- A system.webServer section was added to web.config files provided in the Silverlight project template and samples. [##B993]
- The dependency on System.Windows.Control.Data was removed from IdeaBlade.Core.SL (formerly named IdeaBlade.Util.SL). [##B1000]

IdeaBlade DevForce Release Notes

- The ContainerName attribute generated into edmKey elements in the app.config is now correctly using an assembly name qualifier. [##B1003]
- WPF TraceViewer reference help added. [##B1014]
- The WPFTraceViewer utility now correctly accepts a URL passed via either the command line or to the constructor. [##B1016]
- AssemblyCopyright information updated. [##B1023]
- Object Mapper - Ensure the target project dropdowns are set correctly when a domain model is opened. [##B1031]
- Object Mapper - Fixed a problem in generating the relative path to the EDMX file causing an incorrect path to be written to the IBEDMX file, making it unusable. [##B1032]

Silverlight-Only

- Fixed issue where an exception would occur with a projection fetch while in a disconnected state. [##B954]
- Better handle auto-reconnect logic after connection failures. [##B815]
- Connection failures occurring when testing the Silverlight application within Visual Studio which are caused by the wrong setup project now generate an exception with explanatory text. [##B955]
- The EntityCacheState is now saved in binary format in Silverlight. Depending on the data saved, binary format can provide size reductions between 5 to 25% or more. [##B1027]

Version 4

Version 4.3.0

Summary

Developer facilities for intervening in Gets and Sets of entity properties have been majorly overhauled, reducing generated code side and providing you, the developer, with even more power and flexibility.

Breaking changes

- **App.Config and Generated Code.** Due to changes in the handling of the app.config file(s) and in the way property interceptors are defined and used, **you will need to regenerate your object models.** Having done so you will see a copy of app.config in the DomainModel folder. The AppHelper project will not be removed, but will no longer be used (and can be deleted) unless you have added your own classes thereto.

For the time being, DevForce will continue to support probing at runtime for the app.config in the AppHelper, but this support is deprecated and may be removed in the future.

- **Generate developer partial class files default has changed.** In the Object Mapper, the default value for the “Generate developer partial class files” option has now changed from “true” to “false”. If you left this value at the default previously, it will now be switched off when you open your model in the Object Mapper. We have chosen this more conservative setting in order not to create a nuisance for developers who prefer to locate their developer partial classes in a different project.
- **Custom property interceptors must be converted.** Because of changes in the way property interceptors are defined and used in this version of DevForce, any custom BeforeGet, AfterGet, BeforeSet, AfterSet overrides that you have defined will have to be converted to the new supported format.
- **PropertyName_Reference properties retired.** The PropertyName_Reference properties in the generated code have been retired. These were primarily designed for internal use, but were accessible to developers.
- **Entity Indexers Moved.** Indexers on Entities have been moved to the Entity.EntityAspect. For example, the older syntax...

(Old)

C#	string lastNameAsRetrieved = (string)_currentEmployee[#"LastName", EntityVersion.Original];
VB	Dim lastNameAsRetrieved as String = _ CType(_currentEmployee[#"LastName", EntityVersion.Original], String)

...must now be written as follows:

(New)

C#	string lastNameAsRetrieved = (string)_currentEmployee. EntityAspect [#"LastName", EntityVersion.Original];
VB	Dim lastNameAsRetrieved as String = _

```
CType(_currentEmployee.EntityAspect[#"LastName", EntityVersion.Original], _String)
```

- **IdeaBladeConfig.CopyLocal obsolete.** The IdeaBladeConfig.CopyLocal method has been marked as Obsolete.

New and Improved

AppConfig / AppHelper

In order to synchronize more completely with evolving .NET coding paradigms, we have done away with the AppHelper project. The Object Mapper now generates the app.config file into the DomainModel project. Optionally, it also updates the copy of app.config in the executable project.

For a limited period of time DevForce will continue to probe for app.config in an AppHelper assembly if one is present. However, we encourage all users to retire AppHelper. The role of app.configs in application, and DevForce's work with them, is now documented in Chapter 3 of the Developers Guide, "Hello DevForce", in a section entitled "Understanding the App.Configs".

Object Mapper

You now have the option of specifying where the Object Mapper should generate your domain model. In previous versions, when you created a new domain model, a new class library project and an AppHelper project were always automatically created. The Object Mapper also created a new solution folder and placed all of the new projects, along with the Entity Model project, inside that folder.

Since you now have control over the location of your domain model, before you can successfully save the domain model you must explicitly choose an existing project, or create a new project, to hold it. The Object Mapper will also now create the solution folder only when creating a new class library project. Other project types will not be created within a solution folder. You can of course move your projects to solution folders of your choosing after they've been created. As noted elsewhere, the AppHelper project has been retired, in favor of standard app.config files.

Other Changes to the Object Mapper

- **Object Mapper grid.** In the Object Mapper, we replaced the Infragistics grids with WPF DataGrids based on the October 2008 release of the WPF Toolkit. [##998]
- In the **Object Mapper**, we now **freeze the first column** in the grids so that context is retained when scrolling horizontally. [##955]
- In the **Object Mapper**, we **changed the default setting for the "Generate Developer Classes"** checkbox to unchecked. [##1027]
- In the **Object Mapper**, we removed the **"Output intermediate files"** option from the user interface (project page) and made it a hidden option that can be exercised by editing the .ibedmx file manually and setting the ShouldGenerateIntermediateFiles flag to "true". This option is rarely needed, and only for debugging purposes. [##1034]
- **UpdateFromDomainModelConfig.** This new flag in the app.config file, which must be set manually, permits the developer to control how and whether the copy of app.config in the executable project is

updated by the Object Mapper. The setting defaults to “Ask”. For details, please see the section “Understanding the App.Configs” in the “Hello DevForce” chapter of the Developers Guide.

PropertyInterceptors

A new, much more streamlined facility is provided in this release for intervening in the Get and Set processes for entity properties. This enabled us to reduce considerably the size of DevForce’s generated entity classes, and provides you with even more powerful and flexible mechanisms than were previously available for such interventions. The new facilities are documented in a new chapter of the Developers Guide named “Property Interceptors”.

AsyncSerialTask and AsyncParallelTask

DevForce includes two classes, the *AsyncSerialTask* and the *AsyncParallelTask*, that permit you to define and execute, in series or in parallel, a collection of linked actions. Each method uses a single callback to handle all processing results, and each provides the ability to specify an *ExceptionHandler* to provide a single point of error handling. For details, see the new section “Batching Asynchronous Tasks” in the “Object Persistence” chapter of the *Developers Guide*.

Features

- **New project templates** are now installed into Visual Studio for a BOS Web Application and a BOS Web Site. [#F1016]
- **Additional sample config files** are now installed in the *Program Files\IdeaBlade DevForce WinClient\SampleCode\Deployment\Sample N-tier config files* folder.
- **Changes relating to PropertyInterceptors.** The following two features were added in support of the new Property Interceptor implementation:
 - In the Object Mapper, we removed grid columns and global options relating to Getter Mode and Setter Mode.
 - On generated properties (both simple and navigation), we removed the “virtual”/“Overridable” modifier.

Please see the new chapter, “Property Interceptors”, in the Developer Guide for details.

- A new **EntityPropertyNames nested class** is now generated (as a partial class) within each Entity class. Example:

```
Employee.EntityPropertyNames.LastName // returns "LastName"
```

It has always been possible to avoid hard-coding property names in code using the LINQ PathFor() extension method:

```
Employee.PathFor(e => e.LastName) // returns "LastName"
```

However, a mechanism not involving a method call was needed to return the property names in attributes used to decorate Set and Get interceptors in the new implementation of those. The *EntityPropertyNames* class satisfies this requirement.

- The Winform user **interface** for the **TraceViewer** was replaced with a WPF version.

Learning Units

200 Series] New Learning Units

LiteHausWPF. Sample solution showing a DevForce WinClient app with a WPF user interface.

Verification_WithWinforms. Sample solution illustrating the use of DevForce WinClient's verification facilities in an app with a Winform user interface.

Verification_WithWPF. Sample solution illustrating the use of DevForce WinClient's verification facilities in an app with a Winform user interface.

200 Series] Updated Learning Units

LINQ, Entity SQL, Deployment Tester. Added a copy of the NorthwindEF database (used by the solutions in this unit) to 06 Data folder.

ASP.NET Apps Using AspDatasource. Did a complete walk-through and cleanup of this unit to correct various minor nits. Added new sidebar, "*Maintaining Session State v. Remaining Stateless*", to the tutorial script.

300 Series] New Learning Units

DataSourceKeys. This Learning Unit contains two sample applications:

- **Using DataSourceExtensions.** Shows the basics of working with DataSourceKeys and DataSourceExtensions defined in the app.config file.
- **Custom DataSourceKeyResolver.** Shows how to use dynamically created DataSourceKeys. In the example, connection strings are looked up in a database based upon the application tenant who is logging in.

NorthwindIB Database

The Photos in the Employee table were replaced with higher-resolution versions.

Documentation

Developer's Guide

Hello DevForce Chapter

- Completely revised this chapter to make it current with changes to DevForce WinClient, including the changes in Property Interceptor facilities.
- Added an appendix, "Probing Sequence for the App.Config File", which details how DevForce searches for its configuration information.

Business Object Mapping Chapter

- Added extensive discussion on working Multiple Data Sources using DataSourceKeys to the Object Mapping chapter.
- Added discussion of Push Notification.
- Added material to the Troubleshooting section that addresses the exception "Maximum concurrent users limit met or exceeded".

Property Interceptors Chapter

- New chapter which details re-engineered mechanisms for property interception in DevForce WinClient.

Defect Repairs

The following defects identified in Release 4.2.2 were fixed:

- Added support for the MaxTextLength attribute in EF-WinForms applications. [##856]
- In the Object Mapper, fixed focus issues caused by editing in the grid and immediately clicking in a toolbar button or menu item. [# #910]
- Ensured that a newly added injected type shows up in the list of available base types. [##911]
- Fixed a NullReferenceException that occurred during a save when unchanged entities were encountered. [#B885]
- Fixed a bug in which scalar relations were not identified correctly when inherited types were involved. [#B891]
- DevForce now sets version data when Entity.SetModified is called. [#B893]
- Scalar navigation properties can now be set to null. [#B895]
- Fixed a TracePublisher multi-threading issue. [#B906]
- ByteArrayToString now correctly outputs values between 0x0] 0x0F. [#B921]
- The Object Mapper now warns if an EDMX selected using the Add Entity Model dialog is not in the solution. [#B924]
- DevForce now logs the reason for an invalid license key exception to the debuglog. [#B925]
- Fixed an ArgumentException thrown when performing ID generation after a RejectChanges call. [#B894]

Version 4.2.2

Summary

Two editions of DevForce WinClient are now available that provide you with the opportunity to try out the product without charge.

Other than that, the most significant change for this release is the addition of support for Add() and Remove() on non-scalar navigation properties.

Both of these changes are described below in the section “New and Improved”.

Breaking changes

A reference to the WindowsBase assembly (by Microsoft) is now required in projects that use entities. This reference, if absent, is now added to the DomainModel assembly by the Object Mapper when you regenerate model code. However, you may need to add it manually to your UI project, and other projects that use entities with non-scalar navigation properties. (Let the compiler be your guide.)

The need for this assembly derives from DevForce WinClient’s new support for the Add() and Remove() methods on non-scalar navigation properties (described below under “New and Improved”). To facilitate this support, the RelatedEntityList class used for such properties implements System.Collections.Specialized.INotifyCollectionChanged, which is defined in WindowsBase.

New and Improved

“Free” and “Trial” Versions Now Available

Two editions of DevForce WinClient are now available that provide you with the opportunity to try out the product without charge.

The *Free* edition provides all the features of the DevForce WinClient Professional edition (detailed at <http://www.ideablade.com/DevForceEF/DevForceEFEditionCompare.html>) except that the number of

IdeaBlade DevForce Release Notes

entities that can be generated by the Object Mapper is limited to 15 (instead of being unlimited). The Free edition does not time out after installation: you can use it indefinitely.

The *Trial* edition provides the full Professional Edition feature set, including unlimited entities in the model, but expires 30 days after installation. You can use this edition to test DevForce WinClient with a model of any size.

Note that both of these free editions of DevForce WinClient now support n-tier deployments with an unlimited number of users.

- To install the *Free* edition, enter the word “FREE” when prompted for the license key.
- To install the *Trial* edition, enter the word “TRIAL” when prompted for the license key.

These two passwords are case-insensitive, so any form of capitalization or the lack of it will work.

Support For Developer Express Version 8.3

DevForce WinClient now supports the DXperience 2008 Vol. 3 control suite from Developers Express.

Support For Infragistics Version 8.3

DevForce WinClient now supports the NetAdvantage 2008 Vol. 3 control suite from Infragistics.

Support For Add() and Remove() on Non-Scalar Navigation Properties

RelatedEntityLists returned by all non-scalar navigation properties (e.g., aCustomer.Orders) now support full Add and Remove semantics. A child entity can now be *moved* from one parent to another by adding the child to a different parent.

The support for Add() and Remove() extends to many-to-many relations. Note, however, that in the case of many-to-many relations, a “child” can have many parents so that calling Add() on a child collection will not result in the removal of the specified child from any other parent’s list.

For more detail, see the new material in the Developer Guide *Object Persistence* chapter, in the section entitled, “Adding and Removing Related Objects using Add() and Remove()”.

Enhancements to the NorthwindIB Database

- A new table, EmployeeTerritoryNoPayload, was added to serve as a “payload-free” linking table between Employee and Territory; that is, the table consists entirely of the two foreign key columns *EmployeeId* and *TerritoryID*, which taken together also comprise the primary key.

A pre-existing table, EmployeeTerritory, can also be used to link Employees to Territories. The EmployeeTerritory table does include payload, consisting of an independent, single-column primary key and a RowVersion column used for the detection of concurrency conflicts.

These two styles of linking table [payload-bearing and payload-free] permit exploration of the two different approaches for connecting two entities in a many-to-many relationship in an Entity Data Model. For more detail, see the appendix to the Developer Guide chapter *Business Object Mapping* entitled, “Many-to-Many Associations in the Entity Framework”.

- Photos in Employee table were replaced with higher-resolution versions.

Learning Units

Existing learning units were updated to include newly needed references to the WindowsBase assembly.

Documentation

Developer's Guide

Business Object Mapping Chapter

- A new appendix was added: "Many-to-Many Associations in the Entity Framework".

Object Persistence Chapter

- A new section was added: "Adding and Removing Related Objects using Add() and Remove()".

Installation Guide

- Material was added to describe how to use a Trial key in your installation.
- Two new shortcuts were added to the Start menu under the Documentation section:
 - *Code Samples on the Web*
This shortcut links to a new page on the web site where we will post code samples to help you to perform common important tasks or to improve performance in your DevForce WinClient app.
 - *Training Videos on the Web*
This shortcut links to our web page for DevForce WinClient videos, which cover such topics as Object Mapping, Retrieving Data With LINQ, building a Winform user interface, and working with the Entity Data Model.

Defect Repairs

The following defects discovered by Release 4.2.1 users were fixed:

- The Object Mapper now detects and generates Developer classes in subfolders of the solution when found. [##B788]
- Corrected confusion over Delete and Remove on the part of the EntityGroup EntityChanging events [##B836]
- Corrected a misleading "License Key Violation" message previously generated when DevForce could not find a model in any of the designated probe assemblies. The appropriate message is now returned if unable to find a license key when probing the assemblies in the config file. [## B853]
- Fixed performance problem that occurred in queries for large numbers of related entities. [##B854]
- Corrected an inconsistency in argument Names for OnFetching, OnFetched, OnSaving, OnSaved [## B857]
- EntityManager.GenerateId() now works correctly when called on a sub-typed Entity [##B865]
- Detached related entities are now attached to the correct EntityManager when setting a scalar entity reference. [##B866]
- Error message improved for the exception thrown when Except is used with IEnumerable<T> [##B868]
- Lazy loading is now stopped during pre-fetching via Include. [##B869]
- Object Mapper code generation: EntityRelations are now correctly generated for relations involving sub-types more than two levels deep. [##B871]
- Removed IdeaBlade.Wpf.Controls and IdeaBlade.Wpf.Controls.Design assemblies. [##B872]
- The entity templates were updated in the Create method such that AddToManager is now called from EntityAspect, e.g. anEmployee.EntityAspect.AddToManager(). [##B873]
- Fixed NullReferenceException occurring in ServerConsole.v4.exe under Vista64. [##B877]
- Object Mapper no longer generates the post-build event on class libraries [##B878]

Version 4.2.1

Summary

Perhaps the major change for this release is the delineation of new editions in both the DevForce Classic and DevForce WinClient product lines. IdeaBlade wishes to make DevForce available to a broader set of developers and is, accordingly, making new, lower-cost licenses available in its Standard and Professional editions.

You can find an overview of the product lines at

http://ideablade.com/products_summary.html

Detail on the different editions is available at the following web page:

http://ideablade.com/DevForceEF/DevForceEFEditionCompare_Details.html

In brief, the Standard editions are targeted at two-tier (client-server) work using a single SQL Server database and standard .NET Winform controls (or optionally WPF, if using Visual Studio 2008). The Professional editions support n-tier applications and the Infragistics and Developers Express control suites. The Enterprise editions support the full complement of DevForce capabilities, including multiple databases, disconnected operation, checkpointing, asynchronous queries, verification, IIS integration and SSL encryption, support for custom controls, and the full range of IdeaBlade Customer Support services.

Other changes in this release of DevForce WinClient include a refactoring of the Entity class to improve usability, discussed further under API changes. Many additions were made to the Learning materials, including new videos, tutorials, and sample code solutions. Support for ASP.NET applications was added.

Breaking changes

API Changes

In IdeaBlade.EntityModel.v4

Numerous properties and methods defined in the Entity class and inherited by business entities have been encapsulated within an EntityAspect object in order to reduce clutter when viewing members of business entities (using Intellisense, UI designers, etc.). Now when you do the latter you see only those members directly defined in the specialized class (Customer, Employee, etc.), along with the EntityAspect property. You can drill into the EntityAspect property to get the inherited properties (including EntityKey, EntityManager, EntityMetadata, HasErrors, IsNullEntity, and others) and methods (including Delete, FindRelatedEntities, RemoveFromManager, and others).

One side effect of the refactoring is that methods from Entity that were formerly virtual now are not. The foremost example of this is probably the Delete() method. Instead of overriding this method (to delete child objects before deleting the parent, for example) you should now create a handler for the Changing or Changed event of the EntityGroup to which the targetted type belongs. Here, for example, is code that would delete an Order's line items before deleting the Order:

```
C# private void ConfigureOrderDeleteHandler() {
    EntityGroup orderGroup = _entityManager.GetEntityGroup(typeof(Order));
    orderGroup.EntityChanging +=
        new EventHandler<EntityChangingEventArgs>(orderGroup_EntityChanging);
}

void orderGroup_EntityChanging(object sender, EntityChangingEventArgs e) {
    if (e.Action == EntityAction.Delete) {
```

IdeaBlade DevForce Release Notes

```
Order currentOrder = (Order)e.Entity;
currentOrder.OrderDetails.ToList().ForEach(od => od.EntityAspect.Delete());
}
}
```

For those who prefer, the members of EntityAspect can be made available directly on the business entities as extension methods. To do so, add one of the following statements to the top of your code file:

C# `using IdeaBlade.EntityModel.Extensions.v4;`

VB `Imports IdeaBlade.EntityModel.Extensions.v4`

Once you have done that, code written prior to our EntityAspect refactoring will work as written, except that properties inherited from Entity are implemented as methods (sorry, there is no such thing as an extension property!) and are usually prefixed with the verb Get:

C# (before the refactoring)	<code>aCustomer.EntityManager aCustomer.Delete()</code>
C# (now, using EntityAspect)	<code>aCustomer.EntityAspect.EntityManager aCustomer.EntityAspect.Delete()</code>
C# (now, with extensions made available)	<code>using IdeaBlade.EntityModel.Extensions.v4 ... aCustomer.GetEntityManager() aCustomer.Delete()</code>

NorthwindIB Database

Tables User, UserRole, and Role were added to facilitate illustration of security features available in DevForce.

A complete log of the differences between NorthwindIB and Microsoft's NorthwindEF database (beta 3 version, on which NorthwindIB is based) is installed in the DevForce WinClient installation folder as "NorthwindIB_DifferencesFromEF.txt".

New and Improved

Classes in the Object Mapper's Tree Control Are Now Sorted Alphabetically

EntityManager.DefaultManager Now Returns a Specialized EntityManager

The specialized EntityManager generated by the Object Mapper, whose name you control but whose default name is DomainModelEntityManager, now returns an instance of its own type rather than an instance of the more general EntityManager which must then be cast to the more specialized type. Where formerly you had to do this:

```
C# DomainModelEntityManager _mgr =  
    (DomainModelEntityManager) DomainModelEntityManager.DefaultManager;
```

you can now do this:

```
C# DomainModelEntityManager _mgr = DomainModelEntityManager.DefaultManager;
```

EdmKey Has Two New Properties

IdeaBlade.EntityModel.Edm.v4.EdmKey has two new properties, *DbConnectionString* and *DbProviderName*. You would be likely to work with EdmKey if implementing a custom DataSourceKeyResolver.

Product Installation Wizard

The installation wizard was changed to reflect the new edition breakouts (Standard, Professional, Enterprise).

Learning Units

100 Fundamentals

A new Learning Unit has been added:

- 030 Configuring a Winform

This unit, which includes a step-by-step tutorial, illustrates the basics of working with DevForce WinClient's Winform User Interface facilities, including BindingManagers, BindingManager designers, and the BindableList<T>. See the Winform chapter in the Developers Guide for more information.

200 Intermediate

New Learning Units have been added:

- ASP.NET Apps Using ASPDataSource
- Using a GridBuilder
- Verification

ASP.NET Apps Using AspDataSource includes a four-part tutorial and addresses the use of DevForce WinClient and the ASPDataSource component in web applications.

Using a GridBuilder shows the use of a GridBuilder class to create and configure DataGridViews dynamically in your DevForce app. The GridBuilder gives you code-based control over the details of grid construction, permitting you to establish and enforce standards for grid behavior and appearance throughout your app. It can also be used to construct a grid at runtime from specifications obtained from your user. Be sure to read the ReadMe.txt file included in the Visual Studio solution.

Verification includes a sample app that uses all major types of verifiers available in the DevForce WinClient verification system. See the ReadMe.txt file included in the Visual Studio solution, and the Verification chapter in the Developers Guide, for more information

Documentation

Developer's Guide

Chapter 3, “Hello DevForce”. Added brief documentation of various parts of business object model and where they should be deployed. Full treatment in the deployment chapter.

Chapter 6, “Object Persistence”. In the Entity Caching section, documented the setup and use of the TraceViewer utility for monitoring server activity in a DevForce app.

Online Videos

Videos were added to the IdeaBlade web site on the following topics:

- Retrieving Data With LINQ, the Entity Data Model, and the DevForce Domain Model
- Composing Winforms with UserControls
- Modifying Your Database and Updating Your Entity Data Model

These videos join one on Object Mapping that was previously there. You can find all on the IdeaBlade web site at http://www.ideablade.com/DevForceEF/videos_DevforceEF_fundamentals.html#1.

Installation Guide

The installation guide was updated as follows (organized by chapter):

Installation

Added the URL for Visual Studio 2008 Service Pack 1.

Screen shots from the Installation Wizard were updated to reflect changes to the wizard.

Updating Existing Applications

A screen shot from the Installation Wizard was updated to reflect changes to the wizard.

Documentation and Tools

A new Product Key Updater utility is described.

Defect Repairs

The following defects discovered by Release 4.2.0 users were fixed:

- Includes now operate properly even if they occur elsewhere than at the end of a linq expression [##781]
- StoredProcQuery and WsQuery now have constructor overloads which accept an EntityManager [##800]
- Object Mapper now correctly treats the removal of an EntityModel as a change that should trigger a prompt to the user to save unsaved model changes upon exiting the Object Mapper. [##807]
- EntityManager now correctly throws an exception when encountering a non-invertible Scalar query submitted with the FetchStrategy DataSourceAndCache. Such an operation attempts a union of two scalar values, which is an undefined operation. [##808]
- The Object Mapper now correctly modifies the Entity Data Model when Getter Access for a navigation property is modified. Previously such a change caused the EDM to fail to display. [##817]
- The Object Mapper now correctly generates model code for a property when the access is set to Private. Previously it attempted to make the property virtual, which it does for public properties. [# #818]
- A problem with data retrieval when a navigation property was commented out the navigation property on one side of a relation was fixed. [##819]
- Object Mapper now generates a correct summary comment for navigation properties [##822]

- The Object Mapper previously permitted the user to edit the namespace on the property page for an Entity Data Model; however, since this change did not affected the generated code for the Domain Model, the EDM namespace has been made readonly within the Object Mapper. [##825]
 - Certain non-scalar queries that previously returned a null now return an empty list. [# #827]
 - Avoid casting (and casting errors) with EntityManager.DefaultManager. Every generated DomainModel now has a DefaultManager property which returns the sub-typed EntityManager. [##830]
 - The Object Mapper now sets the ConcurrencyStrategy correctly during code generation. [##831]
 - An unused column in the Object Mapper grid, “Default Verification”, was removed. [##833]
 - Object Mapper now generates EntityRelation code correctly for relations involving sub-types. [##835]
 - EntityGroup EntityChanging events confuse Delete and Remove. [##836]
 - Entity.Clone now sets EntityState as detached and allows dynamic recreation of EntityKey. [##839]
 - DevForce now write the product version number to the debug log. [##843]
 - RelationNames are now namespace-qualified so that conflicts do not occur between similarly named relations from different Entity Data Models participating in a common Domain Model. [##844]
 - Relations involving a subtype are now always translated properly to an EF query. [##845]
 - Default values are now initialized correctly for non-nullable short and byte properties. [##849]
 - A performance issue previously experienced on certain queries involving large numbers of entities with parents was fixed. [# #854]
-
- EntityManager.GetEntityGraph now supports empty spans. [F#699]
 - Valid values are now provided for all required properties upon adding a new entity to an EntityManager [F#934]
 - Service initialization errors are no longer generated if unable to write to the Event Log due to machine security settings. [F#959]

Known Limitations

- Add() and Delete() are not currently supported on navigation properties. This support will be added in the next release of DevForce.

The absence of Add() and Delete() on navigation properties does not stop you from adding related instances to a given entity’s collection of same.

For the case of entities related in 1-to-1 or 1-to-many relationships, related entities can be added (and will therefore subsequently be returned by navigation properties) simply by assigning an instance of the independent entity to the appropriate navigation property of the dependent entity (e.g., assign a Customer to the anOrder.Customer property):

```
C# Customer aCustomer =  
    anEntityManager.Customers  
        .Where(c => c.CompanyName == "Acme Engineering").First();  
Order anOrder = Order.Create(DateTime.Now);  
anOrder.Customer = aCustomer;  
// aCustomer.Orders will now return, among its other orders,  
// the order referenced here as "anOrder"
```

```
VB Dim aCustomer As Customer
    aCustomer = anEntityManager.Customers _
        .Where(c => c.CompanyName == "Acme Engineering").First()
    Dim anOrder As Order
    anOrder = Order.Create(DateTime.Now)
    anOrder.Customer = aCustomer
    ' aCustomer.Orders will now return, among its other orders,
    ' the order referenced here as "anOrder"
```

The above technique can also be used in the case of entities related many-to-many through linking objects that appear in the conceptual model of the Entity Data Model (which is the case for linking entities based on tables that include “payload” (columns in addition to the two foreign keys). For those cases you simply add a new instance of the linking entity and assign the appropriate instances of the many-to-many relatives to its navigation properties.

```
C# // Customers and SalesReps have a many-to-many relationship by virtue
    // or their individual 1-to-many relationships to Order
    Order anOrder = Order.Create(DateTime.Now);
    anOrder.Customer = aCustomer;
    anOrder.SalesRep = aSalesRep;
```

```
VB ' Customers and SalesReps have a many-to-many relationship by virtue
    ' or their individual 1-to-many relationships to Order
    Order anOrder = Order.Create(DateTime.Now)
    anOrder.Customer = aCustomer
    anOrder.SalesRep = aSalesRep
```

In such a scenario, if Customer is to have a SalesReps navigation property, or SalesRep a Customers navigation property, you must manually code these in the Customer and SalesRep developer classes. This will always be the case for entities related many-to-many through linking entities that appear in your conceptual model. (Don’t worry: such navigation properties are easy to write!)

This technique cannot be used in the case of entities related many-to-many through no-payload linking objects because, in that case, no linking type appears in the conceptual model. Current workarounds include the following:

1. Add at least one payload column in your linking table (e.g., a single-column arbitrary primary key, separate from the foreign keys). This will cause the linking entity to appear in both your Entity Data Model and your DevForce Domain Model.
2. Explicitly add the payload-less linking entity to your Entity Data Model and configure associations between it and the endpoint types.

Support will be added in the next release of DevForce WinClient for Add() and Delete() methods on navigation properties, and will extend to entities related 1-to-1, 1-to-many, or many-to-many.

Version 4.2.0

Summary

Breaking changes

API Changes

In IdeaBlade.EntityModel.v4:

- On Entity, properties indicating the state of an Entity have been converted into extension methods on an EntityState enumeration, in order to reduce the Intellisense footprint of Entity. Thus, what you formerly referred to as follows:

C# or VB	<code>aCustomer.IsAdded</code> <code>aCustomer.IsAddedOrModified</code> <code>aCustomer.IsDeleted</code> <code>aCustomer.IsDeletedOrDetached</code> <code>aCustomer.IsDeletedOrModified</code> <code>aCustomer.IsDetached</code> <code>aCustomer.IsModified</code> <code>aCustomer.IsUnchanged</code>
-------------------------	--

... you will now refer to this way:

C# or VB	<code>aCustomer.EntityState.IsAdded()</code> <code>aCustomer.EntityState.IsAddedOrModified()</code> <code>aCustomer.EntityState.IsDeleted()</code> <code>aCustomer.EntityState.IsDeletedOrDetached()</code> <code>aCustomer.EntityState.IsDeletedOrModified()</code> <code>aCustomer.EntityState.IsDetached()</code> <code>aCustomer.EntityState.IsModified()</code> <code>aCustomer.EntityState.IsUnchanged()</code>
-------------------------	--

- The `SaveOptions.ThrowExceptionOnError` property (which defaulted to `True`) has been removed. Now all `SaveChange()` calls throw exceptions in the event of an error unless the `EntityServerError` event is “handled”. See the Developers Guide for details on handling `EntityServerError`.
- `ExecuteQuery` overloads that took an `EntityManager` or `QueryStrategy` have been retired. To run a query using a different `EntityManager` or `QueryStrategy` than the one with which it was defined, use the `With` extension methods. See the Developers Guide for details.
- The `EntityManager` events `ExecuteQueryCompleted` and `InvokeServerMethodCompleted` were removed. These events were used to set up handlers for asynchronous query and `InvokeServerMethod` calls. All asynchronous methods now take a callback as part of their signature and do not use events to signal completion.

IdeaBlade DevForce Release Notes

- The Fetched event on the EntityManager now fires for both synchronous and asynchronous queries. The EntityFetchedEventArgs passed to the handler indicates whether the query was completed synchronously or asynchronously.
- EntityFetchedEventArgs] Fetch results are now available in the Result property of the arguments. There is also now a strongly-typed EntityFetchedEventArgs<T> that is used when an ExecuteQueryAsync<T> is called. These classes now contain Error and Cancelled properties if the query did not complete.
- The Saved event on the EntityManager now fires for both synchronous and asynchronous saves. The EntitySavedEventArgs passed to the handler indicates whether the save was completed synchronously or asynchronously.
- EntitySavedEventArgs] The SaveResult is now available as a property of the arguments. This class now contains Error and Cancelled properties if the save did not complete.
- The ExecuteQueryAsync method on the EntityManager no longer accepts a timer interval to support repeated execution.

NorthwindIB Database

Minor changes were made to NorthwindIB, the database used by many DevForce WinClient tutorials:

- In Employee table, the width of the LastName column was increased from 20 to 30. The width of the FirstName column was increased from 10 to 30.
- In the Customer table, the string-valued primary key was replaced with a uniqueidentifier (GUID) key. The old key is still in the table, renamed to CustomerID_OLD.
- In table Order, the CustomerID column (formerly a foreign key) was renamed to CustomerID_OLD. A new uniqueidentifier CustomerID foreign key column was added. The new column was supplied with data to maintain the existing links between Customers and Orders.
- A ReportsToEmployeeId column was added to table Employee. A recursive relationship was added to the Employee table, using ReportsToEmployeeId and EmployeeID as the linking columns. Values were added to the ReportsToEmployeeId column to establish reporting relationships.

A complete log of the differences between NorthwindIB and Microsoft's NorthwindEF database is installed in the DevForce WinClient installation folder as "NorthwindIB_DifferencesFromEF.txt".

New and Improved

Windows Forms Support

Support for Windows Form user interfaces is now Integrated into the installation executable. DevForce WinClient now:

- Supports binding of DevForce WinClient Entities to all Windows controls currently supported by IdeaBlade DevForce 3.6.3;
- Supports binding of DevForce WinClient Entities to Windows controls from Developer Express and Infragistics;
- Installs a tool for Assembly Binding Redirection that provide support for minor releases of Developer Express and Infragistics controls that are not explicitly targetted by the DevForce dlls for those products;
- Adds a tool for adding a panel for IdeaBlade Windows Forms controls to the ToolBox;

- Includes a Learning Unit, “Composing WinForms with User Controls”, with a sample application showing a Windows Forms front end working with a DevForce WinClient persistence framework.

Methods That Communicate with the Business Object Server Now Have Asynchronous Versions

The EntityManager now supports asynchronous versions of methods which communicate with the BOS. These methods include:

- LoginAsync
- LogoutAsync
- ExecuteQueryAsync
- ExecuteQueryAsync<T>
- SaveChangesAsync
- ForceIdFixupAsync
- RefetchEntitiesAsync
- InvokeServerMethodAsync

The EntityManager supports a hybrid of the event-based asynchronous pattern for this API. We refer to it as a “hybrid” because corresponding events have not been defined for these methods. So instead of subscribing to an event to receive notification about the completion status, you can instead pass a method-specific callback as part of the call. For example:

```
C#
private void RunQuery01() {
    var query = new EntityQuery<Customer>().Where(c => c.Id >= 10);
    int token = 1;
    _entityManager.ExecuteQueryAsync<Customer>(
        query, QueryCompletedCallback, token);
}

private void QueryCompletedCallback(EntityFetchedEventArgs<Customer> args) {
    var resultList = args.Result;
}
```

It’s often simpler to pass a lambda expression for the callback instead of defining a new method. For example:

```
C#
private void RunQuery02() {
    var query = new EntityQuery<Customer>().Where(c => c.Id >= 10);
    int token = 2;
    _entityManager.ExecuteQueryAsync<Customer>(query,
        (args) => {var resultList = args.Result;}, token);
}
```

You can identify this hybrid pattern by the *OperationNameAsync* naming convention.

For those needing additional control over their asynchronous operations the EntityManager also supports the IAsyncResult asynchronous pattern through an explicit implementation of the IEntityManagerAsync

interface. You will need to cast an `EntityManager` to this interface in order to use methods following this pattern. In the `IAAsyncResult` pattern an asynchronous operation is implemented as two methods named **`BeginOperationName`** and **`EndOperationName`** to begin and end the asynchronous operation "`OperationName`". More information on using this interface is available in the DevForce Framework Help reference.

Asynchronous Fulfillment of Navigation Property Queries

DevForce WinClient returns data for navigation properties (such as `Order.Customer` or `Order.OrderDetails`) by issuing queries. Explicit queries in your DevForce WinClient app can be written using the asynchronous method calls detailed above, but control over the fulfillment of navigation properties must be exercised in a different manner.

The `EntityManager` now has a boolean `UseAsyncNavigation` property that can be set to specify that navigation properties should be fulfilled using asynchronous queries.

When reference is made to a navigation property, DevForce WinClient returns either an entity (if the property is scalar) or a `RelatedEntitiesList<T>` (for collection properties). Entities now have an `IsPendingEntity` property;

When `EntityManager.UseAsyncNavigation` is set to true, the entities initially returned for scalar properties, and the `RelatedEntityLists` returned for collection properties, will have a “pending” state until the asynchronous query issued for their fulfillment actually returns data. This state can be diagnosed with one of the following properties:

- `Entity.IsPendingEntity`
- `RelatedEntityList<T>.IsPendingEntityList`

Entities and `RelatedEntityLists` also now have events that fire when the data for pending entities is returned. These are:

- `Entity.PendingEntityResolved`
- `RelatedEntityList<T>.PendingEntityListResolved`

Handlers can be attached to these event to perform actions when the data for pending entities becomes available to your app.

Execute() Extension Method on IEntityQuery

The extension method `Execute()` is now available on most `IEntityQuery` subclasses. Thus, the following two statements are equivalent ways of executing a specified query against a particular `EntityManager em`:

```
C#  
em.ExecuteQuery(query)  
query.With(em).Execute()
```

FirstOrDefaultEntity() Extension Method on IEntityQuery

Another new extension method, `FirstOrDefaultEntity()`, is also now available on most `IEntityQuery` subclasses. This query returns either:

- the first entity in an ordered sequence defined by a query; or (if the sequence contains no elements),
- the `NullEntity` for the targeted type.

Test Mode for the Business Object Server

A “test” mode feature is now available for the Business Object Server (BOS). When the test mode is on, the BOS will support an unlimited number of users for 1 hour, after which point it shuts down. This feature may be useful if you have a limited user license but during development and testing need to support additional logins.

To turn this feature on, add an attribute to the <IdeaBlade.v4> element in your configuration:

```
XML <ideaBlade.v4 version="4.00"
      useDTC="false"
      copyLocal="false"
      loginManagerRequired="false"
      testMode="true"
    >
```

Check the *debuglog.xml* output on your BOS for messages relating to this feature. You will see messages both when test mode starts, and again when the test mode interval has elapsed. After the test mode interval has elapsed, all connection attempts by clients will be refused, and these refusals will be seen in the log also.

After the test mode interval has elapsed, you will need to restart the BOS if you need to continue in this mode.

To turn test mode off, either remove the attribute from the <IdeaBlade.v4> element, or set the value to “false”.

[#Feature #952]

Learning Units

100 Fundamentals

The three-part tutorial series on Object Mapping has been updated to the current release.

The three-part tutorial series on Retrieving Data has also been updated to the current release. An incomplete fourth part has been completed, and a fifth part added.

200 Intermediate

New Learning Units have been added:

- Composing WinForms with UserControls
- Adding and Deleting Entities
- Handling Concurrency Conflicts

Documentation

Developer’s Guide

Chapter 4, “Business Object Mapping”. The main walk-through of the Object Mapper has been reworked to reflect changes and new options in the Object Mapper, including new code generation objects, and facilities for injecting base types into the model.

Chapter 6, “Business Object Persistence”, includes a new section, “Batching Asynchronous Tasks”, that describes the two new classes *AsyncSerialTask* and *AsyncParallelTask*. These classes permit you to define and execute asynchronously, in series or in parallel, a collection of linked actions.

IdeaBlade DevForce Release Notes

Chapter 7, “WinForm User Interfaces”. Extensive documentation of the newly integrated WinForm support added. Chapter renamed from “UI and General Development”.

Chapter 9, “Business Object Server”. This chapter, originally based on material applicable to the version 3.x (DevForce Classic) Business Object Server, was updated to reflect facilities and practices appropriate for the DevForce WinClient Business Object Server.

Chapter 12, “Deployment”. This chapter, originally based on material applicable to the version 3.x (DevForce Classic) Business Object Server, was updated to reflect facilities and deployment practices appropriate for DevForce WinClient.

Installation Guide

The installation guide was updated as follows (organized by chapter):

Preface

A section was added: “Compatibility with Optional Third Party Control Suites”.

Installation

- Discussion of Windows Forms support was added
- In the “After the Install” section, a discussion of the DevForce ToolBox was added.

Updating Existing Applications

A new section on “Updating Third-Party UI Control Suites” was added.

Documentation and Tools

Documentation added for “Assembly Binding Redirector”, “Tool Box Installer”, and “Trace Viewer”.

Troubleshooting

A section, “ToolBox Installer Fails”, was added.

Defect Repairs

The following defects discovered by Release Candidate 1 users were fixed:

- Scalar queries that returned a server-side null failed client-side instead of returning a NullEntity. [# #759]
- The Configuration Editor now accepts a command line argument, and can thus be used within Visual Studio as the default editor for the app.config in your domain model projects. [##773]
- Computed (store-generated) fields are now correctly handled during save processing. [##778]
- Includes did not operate properly if they occurred anywhere except the end of a LINQ expression [##781]
- Span Includes didn’t update entity references in certain polymorphic queries.[# #782]
- Entities of types specified in the SaveOptions.ExcludeFromPostSaveRefetch were left in a modified state after a SaveChanges call. [##795]
- Fixed Object Mapper to add project dependencies to server model projects only when not already present.[##809]
- Fixed Object Mapper to use the \$(SolutionDir) macro in the generated post-build event. [##810]

Known Limitations

- Compiled queries are not yet supported.

IdeaBlade DevForce Release Notes

- Use of non-public access modifiers for types and properties in your model may cause incorrect results or exceptions.

This concludes the Release Notes for DevForce WinClient.