Contents

It's not hard to **migrate an application from DevForce 2010 to the current generation of DevForce**. We'll describe the steps required here.

Notes

Side-by-side installation of DevForce 2010 version 6.1.9 and later, and DevForce version 7.0.2 and later, is supported. If you are using an earlier version of either product you will need to upgrade if you require side-by-side support. See Side-by-side DevForce 2010 installation for more information.

Convert the solution

For every project in your solution using DevForce assemblies, you'll need to re-target the framework and install the appropriate DevForce package(s). You'll also need to upgrade the entity model.

If you haven't previously opened the solution in Visual Studio 2012+, its upgrade wizard will run and generate a migration report.
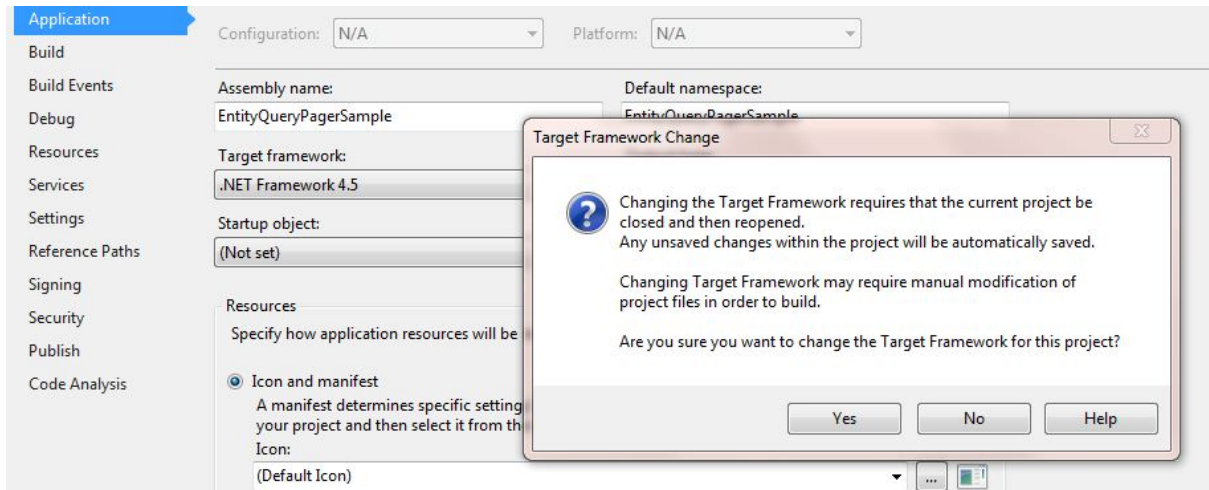
"Clean" the solution first. This will remove existing DF2010 assemblies from your bin folders.

Change the target framework

Re-target Silverlight projects to Silverlight 5.
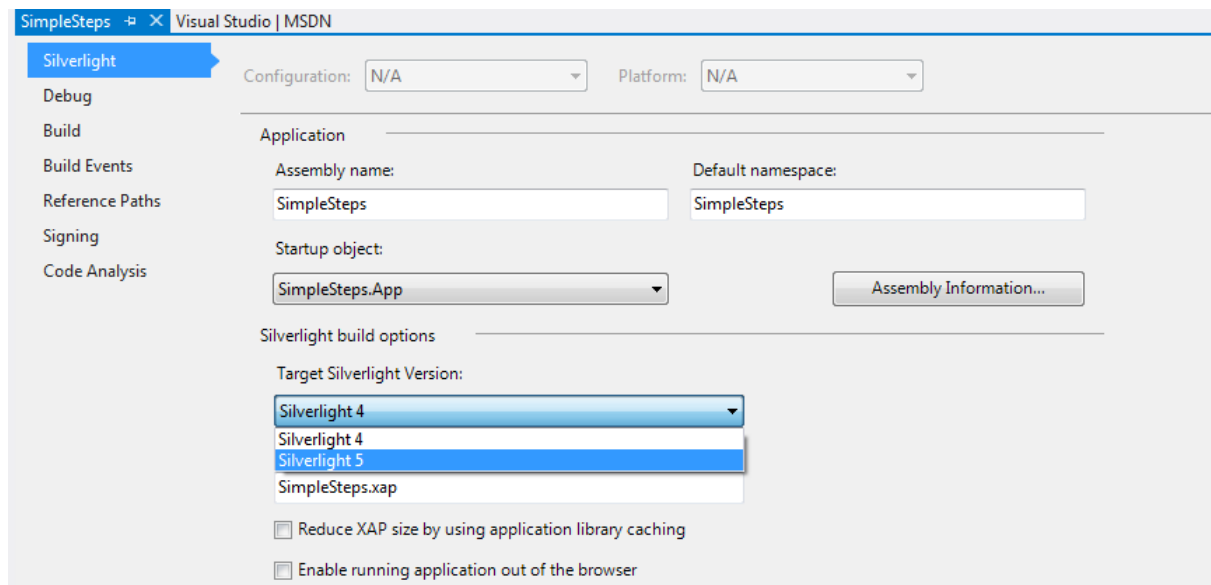
Re-target .NET projects to .NET 4.5+.

Install the DevForce NuGet packages

There are several DevForce NuGet packages, but they aren't all needed for every project.

The packages will not install if you haven't re-targeted your projects to .NET 4.5+ and Silverlight 5.

    Tip: You can install a package to multiple projects at one time. If you install at the solution level, the package manager will prompt for the projects to install to.

All projects using DevForce

Generally **all projects using DevForce** will need the [DevForce Core NuGet package](#). This package installs the "core" DevForce assemblies: IdeaBlade.Core, IdeaBlade.EntityModel, IdeaBlade.Linq and IdeaBlade.Validation.

    The package can be applied to any project type: class libraries, WinForms and WPF client applications, console and Windows Service applications, Silverlight, Windows Store and Universal applications, and web applications.

    If you haven't previously installed either the core package or run the jumpstart installer, additional DevForce tools, including the EDM Designer Extension and DevForce templates, will also be installed.

    It's quickest if you install this package once at the solution level, and select all projects to which it should be applied. For example:
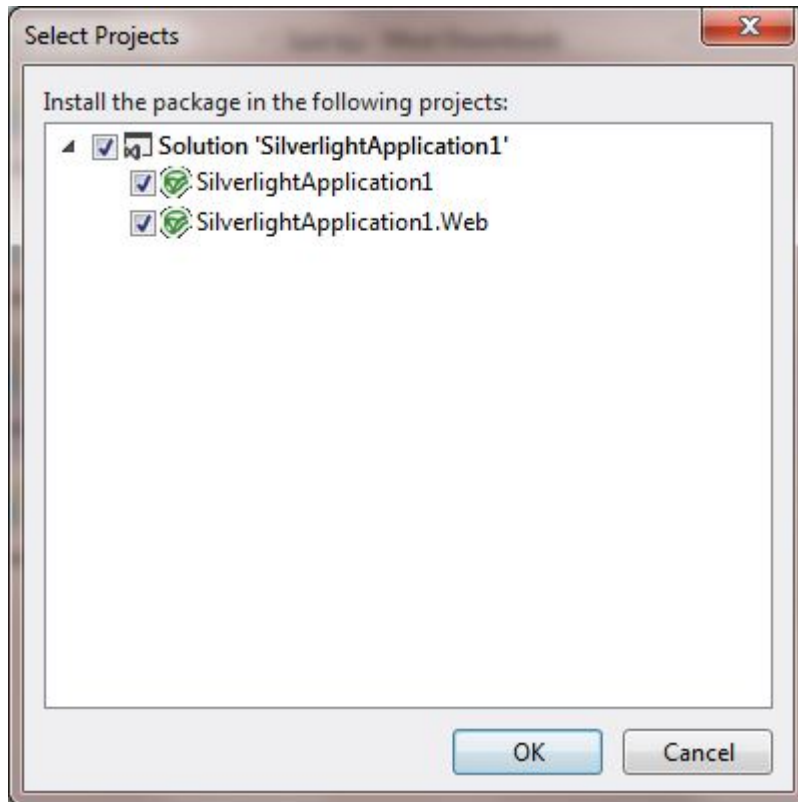
Server projects

For n-tier applications, including Silverlight, the web application server project also needs the DevForce Server NuGet package.

For 2-tier applications, you'll need to install the Server package to your executable project too.

If DevForce 2010 is still installed, check the *IdeaBlade.EntityModel.Web* assembly reference after adding the server NuGet package to a web application project. When DevForce 2010 is installed, this assembly will be in the GAC and the package will not attempt to replace the existing assembly reference. You should remove the assembly reference, and add a reference to the assembly in the packages folder: *..\packages\IdeaBlade.DevForce.Server.7.x.y\tools\lib\net45*.

Code First model projects

You'll need to install the DevForce Code First NuGet package to the project holding your Code First model. If you are working in Silverlight, you should also install the package to your "linked" Silverlight project.

The package will add a reference to IdeaBlade.AOP (or IdeaBlade.AOP.SL for Silverlight), and if the DevForce "core" assemblies are not present the DevForce Core NuGet package is automatically installed too.

The Code First package will also install the EntityFramework NuGet package to add support for Entity Framework 5, and install PostSharp from SharpCrafters.

Both the DevForce and PostSharp packages modify the MSBuild targets for the projects they are installed to. If you peek into the project file you'll see additional imports. For example, a C# project might look something like this:

```
<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
 <Import Project="..\packages\PostSharp.2.1.7.15\tools\PostSharp.targets"
    Condition="Exists('..\packages\PostSharp.2.1.7.15\tools\PostSharp.targets')" />
 <Import Project="..\packages\IdeaBlade.DevForce.Aop.7.0.0-beta1\tools\IdeaBlade.DevForce.Common.targets"
    Condition="Exists('..\packages\IdeaBlade.DevForce.Aop.7.0.0-beta1\tools\IdeaBlade.DevForce.Common.targets')" />
```

The PostSharp targets file ensures that your Code First entities are re-written, while the DevForce targets will fire the *EntityModelMetadataDeploy* task which builds the metadata file for your model.

Upgrade the entity model

An EDMX-based entity model created in DevForce 2010 will be automatically upgraded when you do the following:

- Re-target the project to .NET 4.5 (or above).

- Delete the edmx.tt file. The existing file was generated by DevForce 2010 and contains obsolete assembly references.
- Open the EDMX in the designer.
- Save the model. You should see a new edmx.tt file, along with the *.IB.Designer.* generated code file.

If the DevForce EDM Designer Extension is not installed, or was just installed by the *DevForce Core* package but you have not yet restarted Visual Studio, you will need to restart Visual Studio and ensure the extension is enabled in order to regenerate the model code.

Now for the code!

The asynchronous API has changed quite a bit, for the better. The task-based asynchronous pattern makes coding async calls as easy as coding their synchronous counterparts. There are no more callbacks, operations or completion handlers.   See the topic on asynchronous programming in DevForce for more information.

If you don't wish to make the switch to the new async API now, we offer a compatibility library to help ease the transition. The compat library allows you to keep your existing operation/callback based asynchronous code, with few or no changes. See the topic on using the DevForce Compatibility Pack for more information.

Troubleshooting

1. Check that all DevForce assembly references are for version 7.*. DevForce 2010 assemblies will be version 6.* and are not compatible with the later product.
2. Make sure all DevForce assembly references in all projects are for the same version. DevForce uses a 4 digit version number, for example 7.0.0.0, and the version number must be an exact match among all DevForce assemblies.
3. Remember the server project.

   The DevForce EntityServer always requires two assemblies, in addition to the core assemblies:
   IdeaBlade.EntityModel.Edm
   IdeaBlade.EntityModel.Server

   If the server is hosted by a web application, a third assembly is also required:
   IdeaBlade.EntityModel.Web

   If any of the required assemblies is missing, the server won't start. You'll generally receive an error message stating a specific assembly could not be loaded. On the client, you'll receive the error when you try to construct the EntityManager and connect to the EntityServer.

Known issues

- See the **Release Notes** for additional information about breaking changes and known issues.