IdeaBlade™

Build Better Apps Faster.

DevForce

Universal & WinClient

Installation Guide

Table of Contents

Preface

This Installation Guide describes how to install and upgrade DevForce.

Why the big document?

Installation should be easy. So why do we need so many pages to cover this topic?

Installation *is* easy 95% of the time. Just launch the install executable, answer a few questions, and you're ready to start building your application.

It's that remaining 5% of the time that accounts for most of these pages. InstallShield, Visual Studio 2008, SQL Server – even .NET itself – they all harbor some ugly surprises for the unlucky few. The bulk of this guide will help you recover. When it can't, there is always Customer Support.

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. You can also submit a support case here.

Installation

In this chapter we cover the steps to install DevForce.

- Product Prerequisites
- Pre-installation steps
- Removing earlier versions of DevForce
- Installing DevForce
- Post Installation Tasks

In the "Updating Existing Applications" chapter we explain how to adjust your existing application after installing a new version of DevForce.

Product Prerequisites

Software Prerequisites Summary

- Visual Studio 2008 SP1 or Visual Studio 2010 RC (will not work with Beta 2)
- Silverlight 3 Tools and Silverlight 3 Toolkit (DevForce Universal Only)
- SQL Server 2005, 2008, Express, or other supported database

> Visual Studio Express is not supported as it does not allow add-in components.

Supported Databases

DevForce WinClient object-oriented persistence works with all databases supported by the Microsoft Entity Framework. From reports of those working on providers, these should include all of the following:

- SQL Server
- Oracle
- IBM DB2
- Informix
- MySQL
- PostgreSQL
- SQLite
- Ingres
- Sybase SQL Anywhere
- VistaDB
- Progress

For current information, consult the ADO.NET Entity Framework section of the MSDN Data Platform Developer Center web site (http://msdn.microsoft.com/en-us/data/aa937723.aspx), or try David Sceppa's blog site at http://blogs.msdn.com/sceppa/default.aspx.

Third Party UI Control Suites

DevForce supports binding to the Infragistics and Developer Express control suites. These control suites are not required in order to use DevForce. DevForce generally supports latest versions of these controls within six weeks of their release. Prior to official support from IdeaBlade, you may be able to bind to new versions of these control suites by using assembly binding redirection. This is discussed in the topic "Updating Third-Party UI Control Suites" in this manual.

Pre-installation Checklist

SQL Server

Our tutorials reference the "NorthwindIB" database which may be loaded into any edition of Microsoft SQL Server 2005 or 2008 (including SQL Server Express).

Because many of the Learning Units use NorthwindIB, we recommend that you install at *least* SQL Server 2005 Express on your development machine even if your application will address a database from a different vendor. This, of course, is not a product requirement.

Visual Studio 2008 SP1

If you are using Visual Studio 2008, you will need to install Visual Studio 2008 SP1 before installing DevForce. The download for Visual Studio 2008 SP1 can be found at:

http://www.microsoft.com/downloads/details.aspx?FamilyId=FBEE1648-7106-44A7-9649-6D9F6D58056E&displaylang=en

DevForce supports side-by-side installations of Visual Studio 2008 and Visual Studio 2010 and will install into both environments if detected.

Visual Studio 2010 RTM or RTW

You must have Visual Studio 2010 RTM or RTW installed. The Beta or Release Candidate will not work.

> If you are doing Silverlight development, make sure to select the **Visual Web Developer** feature when you are installing Visual Studio as this is required for all Silverlight development.

If you will be doing Silverlight development, make sure to select the Visual Web Developer feature when you are installing Visual Studio as this is required for Silverlight development.

Silverlight 3 Tools (DevForce Universal Only)

If you are using Visual Studio 2008, you will need to install Silverlight 3 Tools. You do not need to do this if you are running Visual Studio 2010 as it is automatically installed. If you have both environments, you will need to install Silverlight 3 Tools if you want to do any Silverlight development in Visual Studio 2008.

Silverlight 3 Tools contains the Silverlight 3 developer runtime, the Silverlight 3 software development kit, the Silverlight 3 Tools for Visual Studio 2008 SP1, the Silverlight 3 Tools for Visual Web Developer 2008 Express with SP, and some relevant articles from the MSDN knowledge base.

> **You must install the file called *Silverlight3_Tools.exe***. *Silverlight.exe*, by contrast, is just the runtime plug-in and does not contain the Visual Studio SDK needed for development.

You can download Silverlight 3 Tools from here:

http://www.microsoft.com/downloads/details.aspx?familyid=9442b0f2-7465-417a-88f3-5e7b5409e9dd&displaylang=en

Silverlight 3 Toolkit (DevForce Universal Only)

Some of the code samples in the DevForce Learning Resources use controls from the *Silverlight 3 Toolkit* (not to be confused with the Silverlight 3 Tools). The Silverlight Toolkit is:

> …a collection of Silverlight controls, components, and utilities made available outside the normal Silverlight release cycle. It adds new functionality quickly for designers and developers, and provides the community an efficient way to help shape product development by contributing ideas and bug reports. It includes full source code, unit tests, samples and documentation for 26 new controls covering charting, styling, layout, and user input.

You can find the Silverlight 3 Toolkit here:

http://silverlight.codeplex.com/Release/ProjectReleases.aspx?ReleaseId=24246

> We say it again: **the Silverlight 3 Toolkit and the Silverlight 3 Tools are two completely different software packages.** They are only named so similarly in order to separate the really good developers from the rest of the pack.
>
> Be the first on your team to know the difference! ☺

Uninstall Prior Versions of DevForce

If you have installed an earlier version of DevForce (WinClient, EF, or Classic), please uninstall it first.

Removing Earlier Versions of DevForce

> Skip this section if you are installing DevForce for the first time.

Please uninstall prior versions of DevForce WinClient, Silverlight, or Universal before proceeding. DevForce Classic will run side-by-side with this installation and should not be a problem.

If you have manually added any ToolBox items from a previous version of DevForce WinClient, make sure that you have deleted the ToolBox tab and all of the ToolBox items.

- ⇨ Shut down Visual Studio.
- ⇨ Launch "Add or Remove Programs" from the Control Panel.
- ⇨ Find "IdeaBlade DevForce".
- ⇨ Click "Remove" and follow the dialog instructions.

In rare cases InstallShield is unable to uninstall your previous version of DevForce. In this case please refer to the "Uninstall Failure" topic in the "Troubleshooting" chapter.

Replacing the previous NorthwindIB database

Periodically we upgrade the NorthwindIB database that supports the DevForce Learning Units with changes or additions. Check the **Release Notes** to see if we have done so.

> If we have **not** changed the database, you can skip these steps.

If we have updated the database, you should update your copy as well. We won't override your copy so you have to detach it before we will install the new one.

- ⇨ Detach your previous NorthwindIB database.
- ⇨ Delete or relocate the previous NorthwindIB database files.

The new release may come with a revised NorthwindIB database (check the Release Notes for that version).

We recommend that you always keep pace with the most recent tutorial database.

The tutorial database files are "NorthwindIB.MDF" and "NorthwindIB_log.LDF". Be sure to delete or relocate both files after detaching the "NorthwindIB" database.

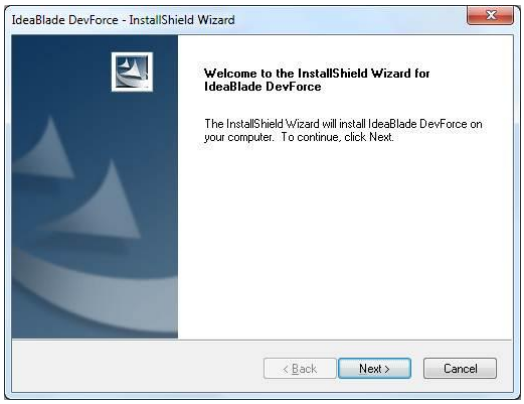
If you forget these steps before you install the new DevForce WinClient, you can upgrade later. Do them first and then run the Database Installer shortcut from the Start Menu.

Installing DevForce

Decompress the zip file and then:

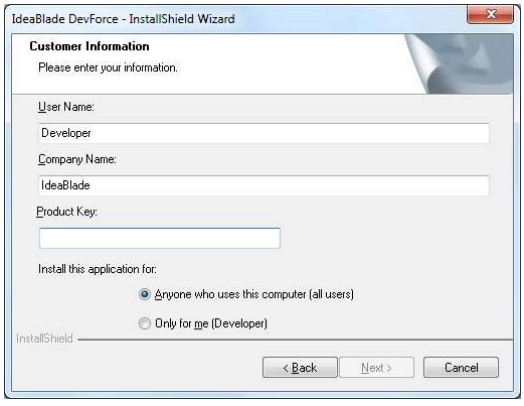
- ⇨ Double-click on the installation executable (e.g., **IdeaBladeDevForce.5.2.6.setup.exe**)

The installer will extract the necessary files and show the welcome screen.



Installation Interview

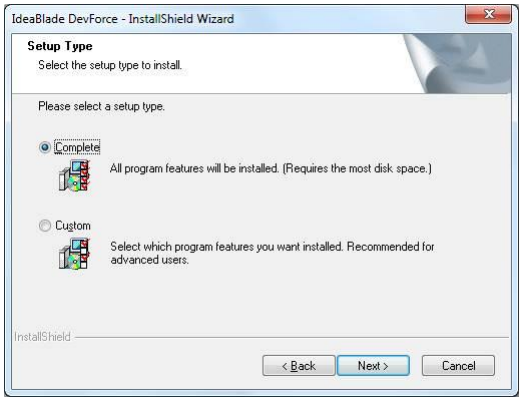
- ⇨ Click Next,
- ⇨ Review and then accept the License Agreement.
- ⇨ Fill in your name, organization, and Product Key



- ⇨ Click [Next]
- ⇨ Assign the destination folder

✓ **Do accept** the suggested destination

- ⇨ Choose either "Complete" or "Custom" installation



Most developers will prefer "Complete".

Start Installation

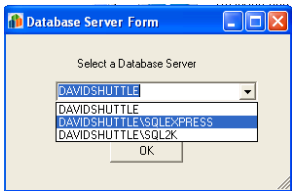
⇨ Click the [Next] button on the "Ready to Install" page.

Aborting the Installation

Try not to abort the installation or un-installation process. If you do, and later have trouble installing or un-installing DevForce, see the "Uninstall Failure" topic in the "Troubleshooting" chapter.

Attaching the NorthwindIB Database

The installer will try to automatically attach the NorthwindIB database used by the tutorials. If you have more than one SQL Server on your system, you'll be prompted to make a choice among your several SQL Servers.

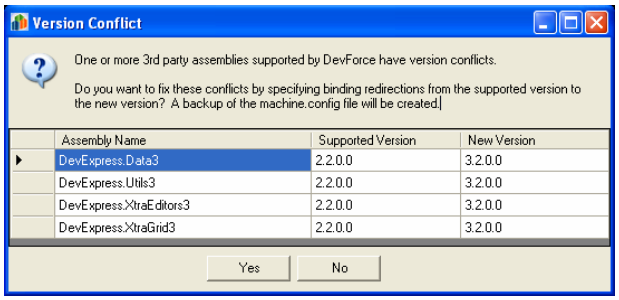


⇨ Select a SQL Server and click [OK].

If the installer fails to attach the database, please see topic in the troubleshooting guide "Trouble Attaching the NorthwindIB Database" or "Manually Installing the NorthwindIB Database".

Assembly Binding Redirection for Third-Party UI Control Suites

If you have chosen to install **Windows Forms Support**, the installation will automatically detect if a third party control suite installed on your machine conflicts with a version supported by this DevForce WinClient release; if so, you'll see a dialog such as this one.

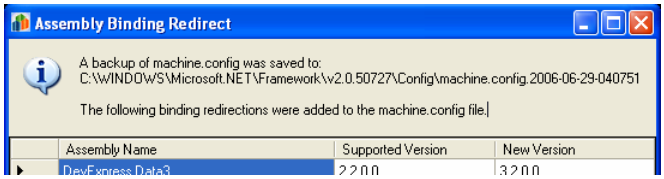


The DevForce supported version may be older than the version of the Third-Party control you installed. The difference is almost always harmless; we'll let you know if we've discovered an incompatibility. Until we've had a chance to catch up, you'll need to tweak some configuration files to tell Visual Studio and your application that it is ok to proceed.

The tool can do some of that tweaking for you but you can opt out by clicking [No]; you'll see a message box before the installation continues.

✓ **Do click [Yes].**

This tool adjusts your "machine.config", the configuration file referenced by Visual Studio when you build your application. The tool creates a backup of your machine.config file before updating it to redirect the DevForce requested version(s) to the new Third-Party tool versions installed on your machine. The tool confirms your choice:



> If there is an error during assembly redirection, the machine.config is restored to its original state and installation continues.

This is only step one of application redirection. It facilitates your application development but you will need to write or modify your application's "application.config" file before deployment.

> See the section on "Assembly Redirection For Developers" for instructions on how to perform assembly redirection manually.

If you have installed a version of DevExpress or Infragistics that requires Assembly Binding Redirection to work properly with the Windows Forms Support feature, you will be prompted for confirmation.

The End

When installation is complete, you should see the DevForce welcome page.

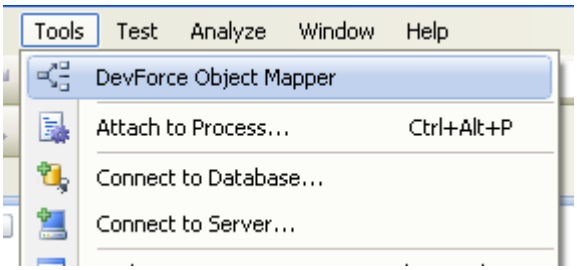
Post-Installation Tasks

Confirming Installation of the Object Mapper

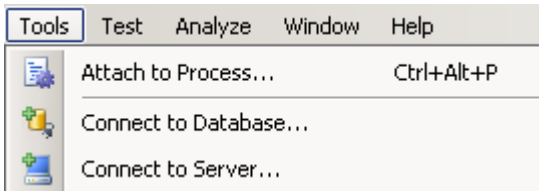
You may want to confirm that DevForce installed properly into Visual Studio. You should be able to launch the Object Mapper from the Visual Studio "Tools" menu.

- ⇨ Launch Visual Studio.
- ⇨ Click the "Tools" menu.

You should see the Object Mapper at, or near, the top of the menu.



Object Mapper present



Object Mapper missing

If the Object Mapper is missing, please see the "Object Mapper Installation Failure" topic in the Troubleshooting chapter.

Updating Existing Applications

Consult the DevForce Release Notes

Should you upgrade to the next release?

We believe it is a good idea to keep up with the current DevForce release. We think it's important to lag no more than one "Feature" release behind.

> We indicate a "Feature" release by changing in the second digit of the DevForce version number (e.g., 4.1 to 4.2); a "Maintenance" release is a change in the third digit (e.g., 4.1.4 to 4.1.5). The fourth digit varies by build and does not typically merit a release note.

We release often – typically once every six weeks. Each release has some new features and some corrections that address the minor defects that crop up in any software.

These changes are meticulously documented in the DevForce Release Notes.

> Access the Release Notes from the "Windows Start Menu | IdeaBlade DevForce | Documentation | Release Notes".

✓ **Read these notes before installing** a new version of DevForce.

Sure they are informative. But more importantly, they tell you about the important, release-specific steps you must take to upgrade your DevForce application from one release to the next.

Upgrading DevForce Versions

Most releases of DevForce are fully backward compatible with earlier versions of DevForce. Upgrading is usually as simple as uninstalling the old version and installing the new one.

We do make breaking changes on occasion although we try very hard to keep this to a minimum. We highlight them in the **DevForce Release Notes**.

You should expect to take the following steps after installing every new release of DevForce.

1. Backup your application files.
2. Refresh your project references to the IdeaBlade assemblies.
3. Follow the upgrade advice in the for this release in the **DevForce Release Notes**

   Many release upgrades require you to re-generate your business model entities
4. Rebuild your business object project(s).

Backup your Application Files

We can hardly emphasize enough the importance of backing up your application frequently and especially before installing development software of any kind.

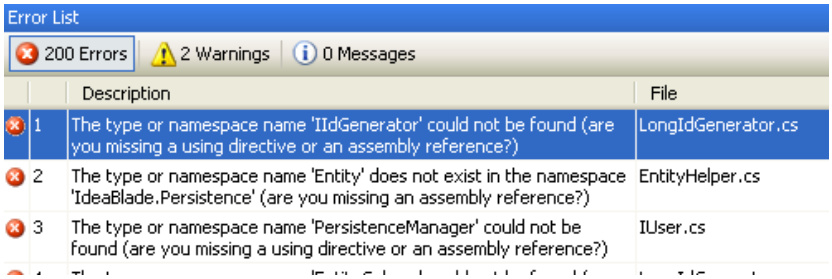
✓ Do use a good automated source control system.

Refresh IdeaBlade References

You must now update the version numbers of the IdeaBlade assembly references in your solution projects.

Background

If you build a pre-existing DevForce application project after installing a new version of DevForce, you may see a flood of errors such as these:



Visual Studio suspects that the problem is a missing reference. Its suspicions are correct – the referenced assemblies may appear to be present in the Project but they are no longer valid.

All IdeaBlade assemblies are strongly-named and are installed in the GAC on the developer's machine.

> They do not go in the GAC of end user machines – and in general they should not be installed there.

All application development assemblies of a new DevForce release receive a new version number – and this number is always the same across the development assemblies.

> The developer tool assemblies such as the Toolbox Installer may have different version numbers.

Consequently, you must update the IdeaBlade assembly references in all of your projects. This need not be an onerous task.

What is the Project's referenced version?

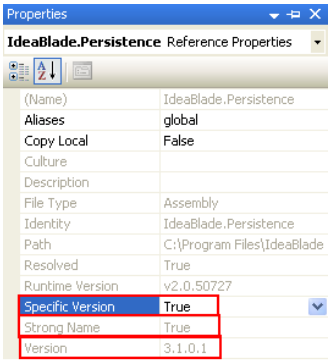
There is no visual clue that the IdeaBlade references are out-of-date.

You could inspect the property sheet of each reference separately.

- ⇨ Right-click a reference
- ⇨ Select "Property" from the context menu

> In a VB.NET project, to see the references in the Solution Explorer, you must activate "Show All" by pressing the button in the Solution Explorer tool bar,
>
> 
>

You will see a property sheet that looks something like this:



The figure highlights three points:

1. The Project requires a specific version
2. This is a strongly named assembly reference
3. The version is 3.1.0.1

> **Beware**: The version number is the version number of the assembly in the GAC, not the version number of the Project reference!

You can easily scratch your head wondering what is wrong. You know that the current DevForce version is 3.1.0.1 and the property sheet seems to confirm that you are referring to that version.

In fact you cannot tell what version the Project is referencing by looking at the property sheet.

Apparently your only clue is that the build thinks there is a missing reference while Solution Explorer says it is there.

The Official Visual Studio Resolution

Visual Studio offers two resolutions, one temporary and one "permanent."

Temporary Workaround

The temporary solution is to disable "Specific Version" checking.

⇨ Select all of the suspect references in Solution Explorer.

⇨ Right-click and select "Property" from the context menu

⇨ Set the "Specific Version" switch to "False" for the selected references.

Now you can recompile and the project will build (unless some other dependent assembly has the same problem).

> This is exactly what we do with our Tutorial projects. The consequences do not seem dire.

"Permanent" Resolution

The proper way is to remove all of these references and recreate them. There is no "refresh" button. You have to

⇨ Select all of the suspect references in Solution Explorer.

⇨ Right-click and select "Remove" from the context menu

⇨ Right-click "References" and select "Add Reference …" from the context menu.

⇨ Hunt for these references again in the ".NET" tab of the "Add Reference" dialog.

Needless to say, this is a time-consuming and error-prone process.

Moreover, you will just have to repeat it with the your next install of DevForce (or any other referenced library).

The Unofficial Fast Alternative

Fortunately, the Project is defined by an XML file named either *ProjectName*.csproj or *ProjectName*.vbproj. You can edit it in any text editor if you're careful.

> Remember to backup

Here is an example:

```
<ItemGroup>
  <Reference Include="IdeaBlade.Persistence, Version=3.0.3.2, Culture=neutral, PublicKey
  <Reference Include="IdeaBlade.Persistence.Rdb, Version=3.0.3.2, Culture=neutral, Public
  <Reference Include="IdeaBlade.Rdb, Version=3.0.3.2, Culture=neutral, PublicKeyToken=28
  <Reference Include="IdeaBlade.Util, Version=3.0.3.2, Culture=neutral, PublicKeyToken=2
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Xml" />
</ItemGroup>
```

The figure highlights the version numbers of the strongly-named IdeaBlade DevForce assemblies. Now we can see that the referenced versions, 3.0.3.2, are not the same as the versions in the GAC which are 3.1.0.1 at this time.

⇨ Update these version numbers

⇨ Save the project file

⇨ Close the solution in Visual Studio (it has cached the prior Project file)

⇨ Re-open the solution.

This is not too painful. With the right tool one could sweep an entire solution full of project files.

Summary

You will have to update your application's references to IdeaBlade assemblies. Your choices are:

- Give up strong version checking.
- Remove and re-add references.
- Edit the references in the Project files.

Regenerate your business object classes

This may not or may not be required by the new release. We indicate in the **DevForce Release Notes** when we think that a release requires regeneration but you should be prepared to regenerate in any case.

> Please do not confuse "*rebuilding*" or "*recompiling*" the model with "*regenerating*" the model.
>
> "*Rebuilding*" is a Visual Studio operation; you will probably have to rebuild in any case.
>
> You can only "***regenerate***" the business model with the DevForce Object Mapper. You will rebuild your solution *after* you've regenerated the model.

The Scary Design View problem

You encountered the scary "Design View" display we described at the opening of this chapter.

⇨ Close this form

- ⇨ Take the actions discussed above:
  - → Refresh your IdeaBlade references
  - → Regenerate and rebuild the business object project
  - → Correct for any breaking changes
- ⇨ Rebuild the solution
- ⇨ Re-open the form in Design View

If you still see the same scary red message, that may still be Ok.

When you opened the solution, it was probably parked in Design View on this Form or UserControl. Visual Studio has cached information about that view – information that is no longer true. Visual Studio won't let go. The remedy is simple.

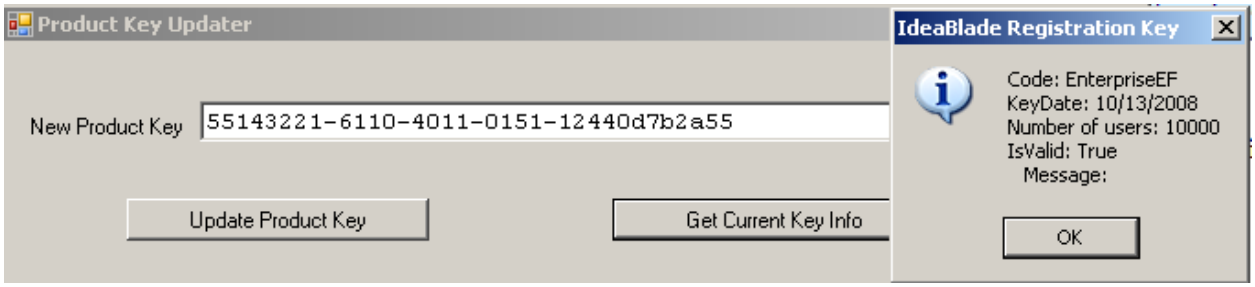
- ⇨ Shut down Visual Studio and re-launch the solution.

Visual Studio will now load its cache with the correct information and the form should display properly in Design View.

Upgrading DevForce Editions

If you are upgrading DevForce Editions, you can accomplish this without having to do a full uninstall/reinstall. The **Product Key Updater** allows you to upgrade your product key and then you can use **Add/Remove Programs** from the Windows Control Panel to install features for your new edition.

You can access the Product Key Updater from the **Start -> IdeaBlade DevForce -> Tools** menu. To update your current key, copy the new key into the **New Product Key** field and press the **Update Product Key** button.



The **Product Key Updater** also allows you to get information about the current key. Just press the **Get Current Key Info** button.

Finally, you should open the Windows Control Panel and use **Add/Remove Programs** to update your DevForce installation with the new features.

Updating Third-Party UI Control Suites

DevForce extends the native .NET data binding architecture. For example, you can bind to nested and dynamic properties of any object as when you bind to the "Name" property of the "State" class when displaying an employee's home address. The bound data property is `HomeAddress.State.Name`.

Some DevForce editions provide special and programmatic support for third party control suites. That means there are vendor-specific "control binders", binding descriptors, binding managers, and visual UI designers.

We build each DevForce release to work with a particular release of the control vendor's product. We do our best to keep up but we may lag a month or more behind.

We cannot simply refer to their control suites by assembly name. We have to refer to them by their strong names. We can only certify our product to work with the controls that we've tested. Thus, our DLLs are locked to specific versions of their DLLs.

Third-Party control updates can break our code but most of the time the changes are transparent to DevForce. Our focus is on the mechanics of binding to their controls. We usually are indifferent to the vendor's improvements and repairs to control styles and behavior.

You, on the other hand, may want the latest control suite update – and want it right now because your application is tripping over a bug or infelicity. What to do?

Fortunately, .NET offers a technique called "Assembly Redirection". With redirection you get to say, in effect, "when the application asks for 3.2.1.3 of this DLL, please use the 3.2.2.0 version instead." If .NET can find that version, your application can continue.

While we cannot guarantee that our code will work with this alternate version it is certainly worth a try; Customer Support may be able to confirm that other customers are doing just fine with the vendor's release.

Assembly Redirection For Client Applications

Your client applications ship with the control vendor's client libraries. You've just downloaded and referenced the vendor's latest release.

You don't want to ship both the current and the DevForce authorized versions of these libraries – even if that is possible – because these are multi-megabyte libraries and installing two sets makes for an overweight application.

You will fix your user interface Project references to point to the current vendor's release, perhaps using the same techniques discussed in the section "Refresh IdeaBlade References " above. Your own code will refer to this new version henceforth.

> Write down the DevForce referenced version number first. You will need to know it when you write your re-direction instructions.

Meanwhile, the DevForce assemblies are pointing to an older version. Let's re-direct them to the new version.

Add an Application Configuration File

You can add a .NET application configuration file to your StartUp Project manually.

> Visual Studio may be able to generate this file for you. These instructions can guide you when you find yourself writing this file manually.

⇨ Select your StartUp Project in Solution Explorer

⇨ Right-click and select "Add ► New Item …" from the context menu.

⇨ Find and select the "Application Configuration File" template.

⇨ Click [Add], thus accepting the suggested name, "App.config".

> If you already have one, open it up.

The tab reveals a .NET XML file named "App.config." It looks like this if it is brand new:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
</configuration>
```

Add the <runtime /> XML

Enter the following XML between the <configuration /> tags.

```
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    </ assemblyBinding>
  </runtime>
```

> If your App.config already has these tags, don't add them again.

Find the Third-Party Control References

⇨ Open the StartUp Project's project file in a text editor.

A Project file ends in either a .csproj or .vbproj extension.

⇨ Find the references to the Third-Party control libraries

For the DeveloperExpress suite they might look like this:

```
    <Reference Include="DevExpress.Data3, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=79868b8147b5eae4" />
    <Reference Include="DevExpress.Utils3, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=79868b8147b5eae4" />
    <Reference Include="DevExpress.XtraEditors3, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=79868b8147b5eae4" />
    <Reference Include="DevExpress.XtraGrid3, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=79868b8147b5eae4" />
```

⇨ Copy these lines to the clipboard

Add Assembly Redirect lines

⇨ Paste these lines between the <assemblyBinding /> tags.

⇨ Tweak each line into four tags as in this example so that …

```
<Reference Include="DevExpress.Data3, Version=3.2.0.0, Culture=neutral,
PublicKeyToken=79868b8147b5eae4" />
```

… becomes

```
<dependentAssembly>
 <assemblyIdentity name="DevExpress.Data3" publicKeyToken="79868b8147b5eae4" />
 <bindingRedirect oldVersion="3.2.0.0" newVersion="3.2.0.2"/>
</dependentAssembly>
```

> Be sure to confirm both the old version number and the new version number.

The resulting file looks like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <runtime>
   <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
     <assemblyIdentity name="DevExpress.Data3" publicKeyToken="79868b8147b5eae4" />
     <bindingRedirect oldVersion="3.2.0.0" newVersion="3.2.0.2"/>
    </dependentAssembly>
    <dependentAssembly>
     <assemblyIdentity name="DevExpress.Utils3" publicKeyToken="79868b8147b5eae4" />
     <bindingRedirect oldVersion="3.2.0.0" newVersion="3.2.0.2"/>
    </dependentAssembly>
    <dependentAssembly>
     <assemblyIdentity name="DevExpress.XtraEditors3" publicKeyToken="79868b8147b5eae4" />
     <bindingRedirect oldVersion="3.2.0.0" newVersion="3.2.0.2"/>
    </dependentAssembly>
    <dependentAssembly>
     <assemblyIdentity name="DevExpress.XtraGrid3" publicKeyToken="79868b8147b5eae4" />
     <bindingRedirect oldVersion="3.2.0.0" newVersion="3.2.0.2"/>
    </dependentAssembly>
   </assemblyBinding>
  </runtime>
</configuration>
```

Confirm those version numbers. The vendor may not use the same version numbers for every library.

⇨ Save the `App.config` file.

What happens to the Application Configuration File?

When you compile (build) your application, Visual Studio copies this application file into the executable directory giving it the full name of the executable plus the extension, `.config`. Thus, if your application creates an executable assembly called `MyApp.exe` in the `..\bin\debug` directory, you'll find a file nearby called `MyApp.exe.config`.

You can learn more about application configuration files from Microsoft and from books, articles, and the web.

Re-directing the next time

In general there is no harm in having unused `<bindingRedirect />` tags in your `App.config`. These re-directions become superfluous as you migrate to the next version of DevForce and all libraries are back in sync.

On the other hand, the vendor may issue yet another release (say, "3.2.0.**4**") before we catch up. You will want to bump to the next version. That's easily done by using the "range" notation for the old version as shown:

```
  <dependentAssembly>
    <assemblyIdentity name="DevExpress.XtraGrid3" publicKeyToken="79868b8147b5eae4" />
    <bindingRedirect oldVersion="3.2.0.0-3.2.0.4" newVersion="3.2.0.4"
  </dependentAssembly>
```

Assembly Redirection For Developers

If you are using DevForce visual UI designers to compose your Third-Party controls, you could run into version mismatch problems during development.

We must apply a similar assembly redirection to your `Machine.config`. `Machine.config` is a file that does for your development environment what `App.config` does for your application.

The DevForce Assembly Binding Redirector Tool can do this for you. You can access this tool from the Start -> IdeaBlade DevForce -> Tools menu.

`Machine.config` is located at `C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\CONFIG` on most machines.

Everyone has a `Machine.config` and it has a significant amount of material in it. We're going to add our redirection tags to it.

- ⇨ Copy everything in the `App.config` between and including the `<runtime> </runtime>` tags.
- ⇨ Open `Machine.config` in a text editor.
- ⇨ Search for the text within these quotes: "<runtime"

> If you don't find the text, "<runtime", scroll to the bottom just above the `</configuration>` tag.

- ⇨ Paste in the assembly redirection tags.

> If the search found "`<runtime />`", we will replace it with the XML we copied to the clipboard.
>
> It there was already a pair of `<runtime> </runtime>` tags with other redirections, merge the copied XML with those pre-existing redirections as appropriate.

- ⇨ Save and close the file.

In general there is no harm in having unused `<bindingRedirect />` tags in your `Machine.config`. These re-directions become superfluous as you migrate to the next version of DevForce and all libraries are back in sync.

We regard the designer classes associated with the domain model as DevForce internal classes and feel empowered to modify their implementation from release to release without bothering you about it … except for this one thing: we may ask you to regenerate them with the Object Mapper.

> Please do not confuse "*rebuilding*" or "*recompiling*" the model with "*regenerating*" the model.
>
> "*Rebuilding*" is a Visual Studio operation; you will probably have to rebuild in any case.
>
> You can only "***regenerate***" the business model with the DevForce WinClient Object Mapper. You will rebuild your solution *after* you've regenerated the model.

Troubleshooting

Identifying Your DevForce Version

From within Visual Studio

An easy way to identify your DevForce version if you're already in Visual Studio is to look at the "About" box of the "DevForce Object Mapper".

- ⇨ Launch the Object Mapper from the Visual Studio Tools menu.
- ⇨ Select "Help ► About" from the top menu.

Of course you can't do this if you're having trouble launching the Object Mapper.

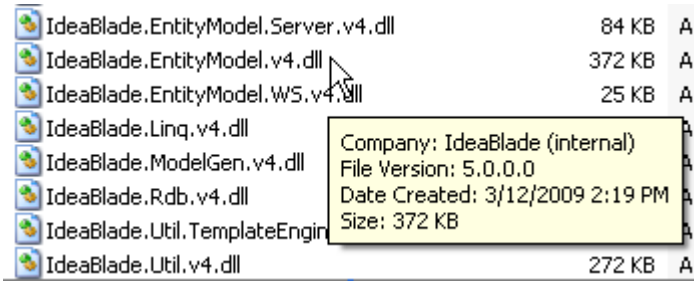
You can view the version number by examining the property sheet of any referenced "IdeaBlade" assembly.

- ⇨ Open the Solution Explorer [Ctrl+W+S].
- ⇨ Open the References folder of your Model or UI project (VB developers must press the "Show All" button first).
- ⇨ Select an IdeaBlade assembly.
- ⇨ Open its property sheet [Ctrl+W+P].
- ⇨ Examine the "Version" property near the bottom.

Outside Visual Studio

A handy and easy alternative is to examine one of the DevForce components in the installation directory, which is typically located at C:\Program Files\IdeaBlade DevForce\ .

- ⇨ Hover the mouse over `IdeaBlade.EntityModel.v4.dll`



Uninstall Failure

General Failure

This can occur if an installation or un-installation was aborted.

First, try deleting any files in the "Program Files/InstallShield Installation Information" folder.

Finally, Microsoft provides a tool, the "Windows Installer Cleanup Utility" that will uninstall just about anything:

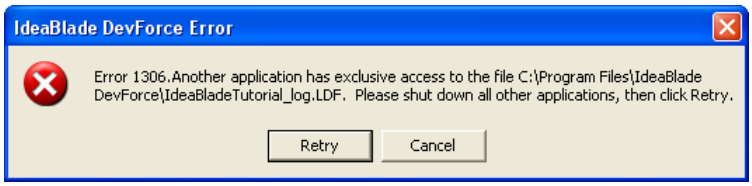
http://support.microsoft.com/kb/290301

> Microsoft often reorganizes their site so if this link fails, search the web for "Windows Installer Cleanup Utility".

After running this tool, if you are trying to clear your machine of DevForce altogether, you might try installing a fresh copy of DevForce and then un-installing it immediately afterwards; this should pick up most if not all of the files from earlier installs.

Blocked by NorthwindIB Database

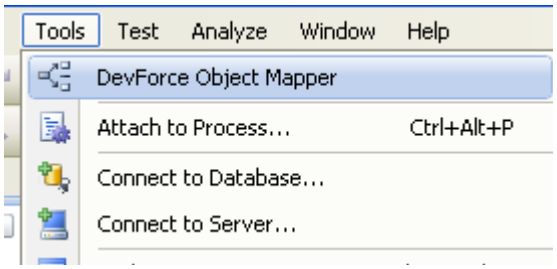
During uninstall you may see this message:



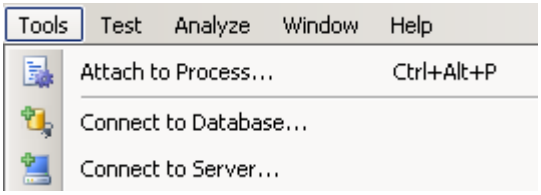
You don't want to cancel because that terminates uninstallation. The work-around is easy.

- ⇨ Launch the SQL Server Management Console
- ⇨ Find the NorthwindIB database
- ⇨ Right-click and select "Tasks ►Take Offline" from the context menu.
- ⇨ Return to the "IdeaBlade DevForce Error" dialog and click [Retry].
- ⇨ After the uninstall completes, return the SQL Server Management Console
- ⇨ Right-click and select "Tasks ►Bring Online" from the context menu.

Object Mapper Installation Failure



Object Mapper present



Object Mapper missing

The Object Mapper should appear as the first choice in the Visual Studio Tools menu. If you don't see the Object Mapper, the following sections will show you how to put it there manually.

> The Object Mapper will not install in Visual Studio Express because it does not support plug-in components.

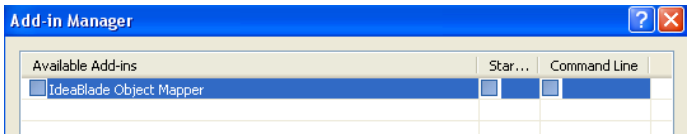
Manual Repair of the Missing Object Mapper

The usual cause of a missing Object Mapper is that Visual Studio has a pointer to a "bad" Object Mapper Add-in.

Check for "bad" Add-in

- ⇨ Launch Visual Studio
- ⇨ Select Tools ► Add-in Manager … from the menu.

Add-in Manager probably shows that the DevForce Object Mapper is present as in the following:



> If you don't see the Object Mapper listed, see the section entitled "No Object Mapper in Add-In Manager", below.

Visual Studio "remembers" the last time it tried to use this Add-in and it can't load what it remembers.

Copying a new version of the Object Mapper Add-in on top of the old copy won't work. Visual Studio won't try to load it again until we make it forget about this Add-in.

The trick is to

- Rename the `IdeaBlade.DevTools.OM.AddIn.AddIn` file temporarily
- Convince Visual Studio that it is gone
- Restore the Add-in file name
- Cause Visual Studio to discover it anew

The AddIns Directory

Visual Studio looks for an XML file in its "AddIns" directory.

That directory varies depending upon your edition of Visual Studio and how you installed it.

The preferred location under WindowsXP is:

C:\Documents and Settings\All Users\Application Data\Microsoft\MSEnvShared\AddIns

> For the All Users location, you must have the Show Hidden Files and Folders option turned on to see the Application Data folder.

Under Windows Vista and Windows 7, the preferred location is

\ProgramData\Microsoft\MSEnvShared\AddIns

on the boot drive.

Alternatively, the directory may be in the user's directory space at

%HOMEPATH%\My Documents\Visual Studio 8\AddIns

> You can copy and paste this into Windows Explorer. Realize that this directory may not exist.

Jiggle the Object Mapper AddIn file

We can proceed now that we know where Visual Studio keeps its Add-in information.

- ⇨ Exit Visual Studio.
- ⇨ Navigate to the Visual Studio "AddIns" directory

- ⇨ Rename `IdeaBlade.DevTools.OM.AddIn.AddIn` to `IdeaBlade.DevTools.OM.AddIn.AddInX`

> Note: it is important that you rename the *extension* on the file. Otherwise Visual Studio will continue to find and use the file.

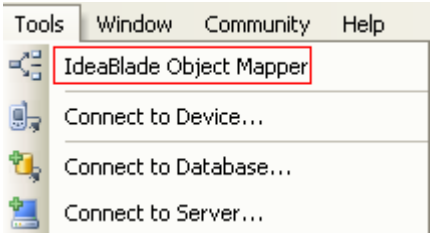
- ⇨ Confirm for Windows that it is OK to change the file extension.
- ⇨ Launch Visual Studio.
- ⇨ Select Tools ► Add-in Manager … from the menu.
- ⇨ Verify that "DevForce Object Mapper" is no longer listed.

> If you still see the Object Mapper listed, you won't be able to revive the Object Mapper. There is some other reference to it somewhere. If you checked both of the potential locations listed above and still can't find it, please contact Customer Support.

- ⇨ Exit Visual Studio.
- ⇨ Restore `IdeaBlade.DevTools.OM.AddInX` back to `IdeaBlade.DevTools.OM.AddIn.AddIn`

That should do it. Now we verify that the Object Mapper is installed.

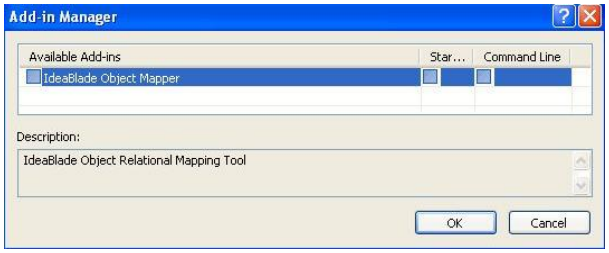
- ⇨ Launch Visual Studio again.
- ⇨ Look for "IdeaBlade Object Mapper" at the top of the Tools menu.



Still can't make it appear?

You've tried the Object Mapper Installer, you've jiggled the AddIn file, and it still won't appear in the Tools menu.

You can see the Object Mapper in the Add-in Manager; it just won't appear in the Tools menu.



Perhaps you customized your installation of Visual Studio. Excluding VB or C# will cause this behavior.

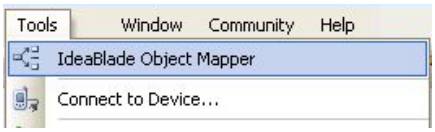
Try the following:

- ⇨ Open the Visual Studio Tools menu.
- ⇨ Select "Customize…".

⇨ Select “AddIns” on the Commands tab. You should see IdeaBlade Object Mapper in the Commands window.

⇨ Select and drag the “IdeaBlade Object Mapper” onto the Tools Menu just above "Connect to Device.

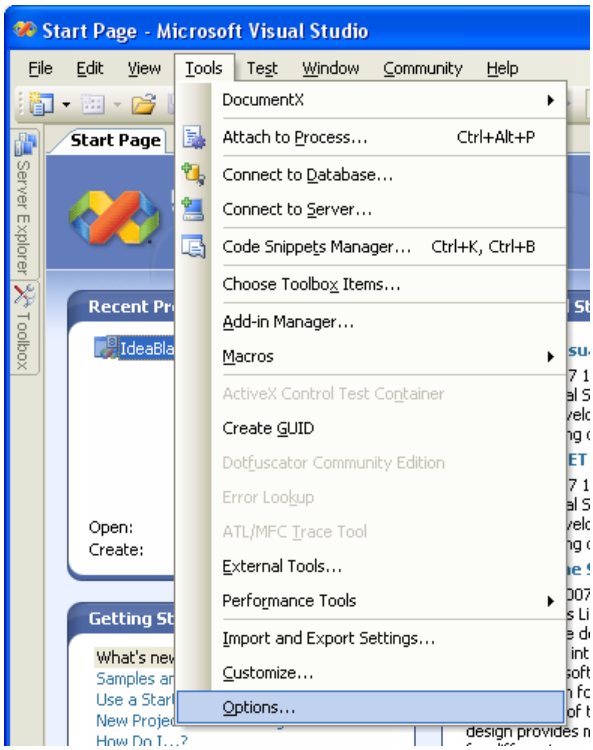
Before drop



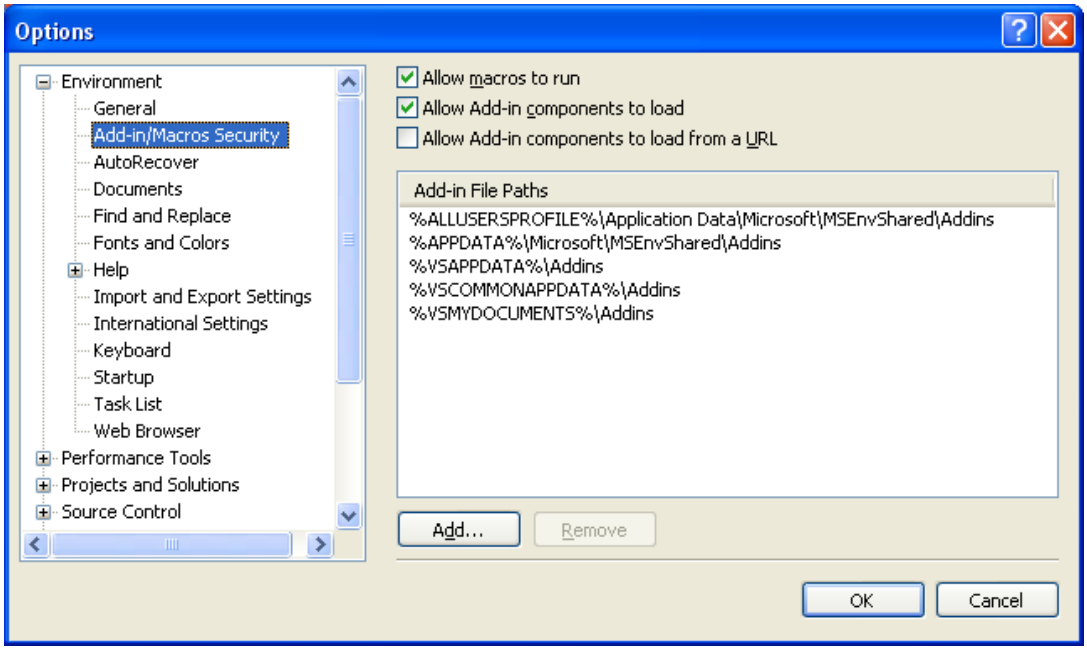
After drop

No Object Mapper in Add-In Manager

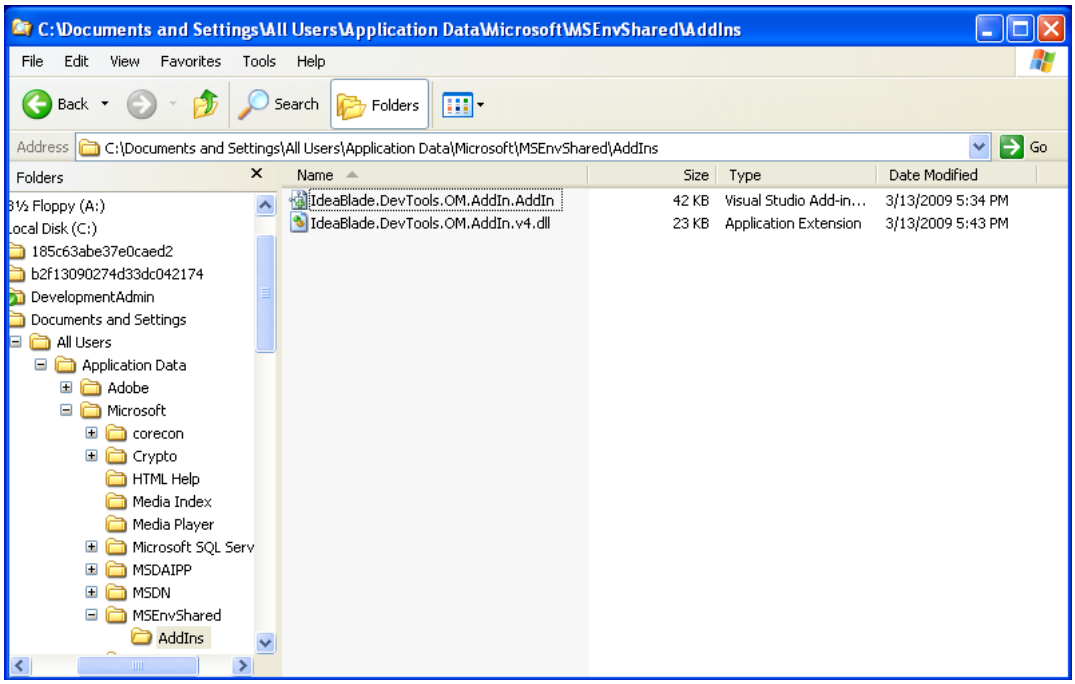
If the Add-In does not show up in the Add-In Manager, it could be that the Add-In path is missing from Visual Studio. To check this, open Visual Studio and select Tools, then Options:



The **Options** window shown below will open. Select Add-In/Macros Security.



Click on the Add button. Navigate to the location where the files IdeaBlade.DevTools.OM.AddIn.AddIn and IdeaBlade.DevTools.OM.AddIn.v4.dll are installed.



Consult the section entitled "The AddInsDirectory", above, for information on where these files are installed by default. Example of default location in Windows XP show above.

Click OK to complete the operation.

This situation may particularly apply to installations of the operating system or Visual Studio that are localized to languages other than English. It may also apply to certain installations on the Windows Vista operating system or to certain makes or models of laptop computers.

Consult this link on the DevForce Forums for more information:

http://www.ideablade.com/forum/forum_posts.asp?TID=25

Trouble Attaching the NorthwindIB Database

The DevForce installer tries to automatically attach the NorthwindIB database so that you will be able to run the tutorials.

If it did not succeed, there could be several possible reasons:

- The installer cannot detect an installed version of MS SQL Server.
- SQL Server is stopped or paused.
- You can't login to SQL Sever
- The installer detects a pre-existing attached copy of the tutorial database.
- There are multiple instances or a named instance of SQL Server.
- The installer cannot attach the tutorial database for some other reason.

> The good news is that DevForce installation can complete successfully even without the NorthwindIB database. The less good news is that if the tutorial database the connection strings in the Tutorials will not be adjusted if you decide to attach the tutorial database manually later.
>
> You can address both problems by running the DataBase Installer from the "Start Menu ► IdeaBlade DevForce ► Tools.

Installer cannot detect MS SQL Server

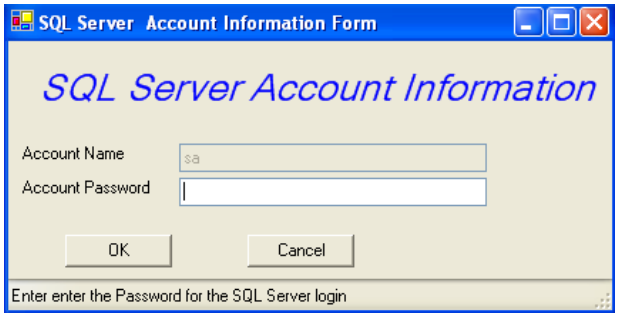
You can continue without attaching the tutorial database. If you later install SQL Server you can run the DevForce Database Installer from Start Menu ► IdeaBlade DevForce ►Tools ►Database Installer.

MS SQL Server is paused or stopped

Start SQL Server and relaunch the DevForst installation. You can also run the DevForce Database Installer from Start Menu ► IdeaBlade DevForce ►Tools ►Database Installer.

SQL Server log-in failure

The installer tries to log in to the master database using Windows Integrated Security. If it cannot, the installer will ask you to log in to SQL Server via the following form.



Notice that the Account Name field is pre-filled with the "sa" (System Administrator) account and that you cannot change that account.

- ⇨ Enter the "sa" account password and press [OK].

If you cannot log in, you can run the DevForce Database Installer at a later date from Start Menu ► IdeaBlade DevForce ►Tools ►Database Installer.

> There are consequences if you can't log-in to the master database either through Windows Integrated Security or via the "sa" account. The Learning Units rely upon a connection string to this database. Each Learning Unit has its own copy of that string.
>
> The installer tries to adjust all of the string copies during the installation process. If the database installation process fails, you will likely have to correct each string manually.
>
> You will find these strings in each Learning Unit's App.Config and in each domain model (.ibedmx) file. You'll learn about these files and how to adjust them in the Developers Guide.

Blocked by pre-existing NorthwindIB database

The installer guards against overwriting a pre-existing NorthwindIB database to avoid destroying any data. If the NorthwindIB database is already attached, or if it still resides in the target database file directory, the installer will silently continue.

This means that if you want InstallShield to install a fresh copy of the NorthwindIB database, you must do both of the following before you install DevForce.

- ⇨ Detach the database
- ⇨ Delete (or relocate) the database files.

> The tutorial database files are "NorthwindIB_Data.MDF" and "NorthwindIB_Log.LDF". Be sure to delete or relocate both files after detaching the "IdeaBlade Tutorial" database.
>
> If you forget these steps before you install the new DevForce, you can upgrade later. Do them first and then run the Database Installer shortcut from the Start Menu.

Manually Installing the NorthwindIB Database

If you have SQL Server running on the system where you install DevForce, we attempt to install the NorthwindIB database automatically. If the Database Installer fails, you can install the `.mdf` and `.log` files manually. If you would like to try this, but do not know how, please contact Customer Support and someone can walk you through the process.

> The Database Installer refreshes the connection strings in the Tutorials so that they work properly with your copy of the database.
>
> These manual installation instructions don't do that so you may have to refresh them yourself. It's easy. Just run the Database Installer again *after* you've attached manually. It will recognize that the database is attached and proceed to refresh the tutorials.

Replacing the previous NorthwindIB database

Periodically we upgrade the NorthwindIB database that supports the DevForce Learning Units with changes or additions. Check the **Release Notes** to see if we have done so.

> If we have **not** changed the database, you can skip these steps.

If we have updated the database, you should update your copy as well. We won't overwrite your copy so you have to detach it before we will install the new one.

- ⇨ Detach your previous NorthwindIB database.
- ⇨ Delete or relocate the previous NorthwindIB database files.

The new release may come with a revised NorthwindIB database (check the Release Notes for that version).

We recommend that you always keep pace with the most recent tutorial database.

> The tutorial database files are "NorthwindIB.MDF" and "NorthwindIB_log.LDF". Be sure to delete or relocate both files after detaching the "NorthwindIB" database.
>
> If you forget these steps before you install the new DevForce, you can upgrade later. Do them first and then run the Database Installer shortcut from the Start Menu.

InstallShield Wizard was Interrupted

If you get a message saying that the "wizard was interrupted", you have probably run into a Windows SP2 operating system security issue. To fix this problem, read the Macrovision Knowledge Base article at:

http://support.installshield.com/kb/view.asp?articleid=Q111303

Assembly Reference Listed but Not Found by Visual Studio

This is a bizarre artifact of Visual Studio 2008 that only occurs for some projects after you've upgraded versions of software and / or upgraded Visual Studio or .NET itself.

When this happens it is easy to address.

- ⇨ Remove the reference from your project.
- ⇨ Re-add the reference.
- ⇨ Re-build the project.
- ⇨ Remember to set "Specific Version" to "false".

Performance Issues

Slow performance inside Visual Studio

Many developers are astonished to discover that database applications that performed crisply in earlier versions of Visual Studio suddenly degrade to unacceptable speeds in Visual Studio 2008.

The good news is that good performance returns when you run the application ***outside*** of the Visual Studio debugger.

✓ **Consider** trying your code outside of Visual Studio.

The Visual Studio debug configuration executable can usually be found in a "\bin" directory under the directory holding the application's "StartUp Project". Look there for the application with the ".exe" extension.

Run that executable and you should see a many-fold improvement.

Fortunately, ***your*** customers will run the executable, not inside Visual Studio.

Poor performance with Visual Source Safe

If you are using Microsoft Visual Source Safe (VSS) you should be aware of the following issues:

1. It is very sensitive to network latency and performs poorly over wide area networks.
2. It substantially slows Visual Studio's ability to manage individual project files.

Many of our customers prefer CVS and Subversion.

Other Performance Concerns

We know good performance is critical to your application. DevForce performs extremely well even with fetches of large amounts of data.

In your pre-DevForce days you would have paid close attention to your hand-coded SQL or stored procedures when reading thousands of records. DevForce makes data retrieval so easy that we tend to forget it has to generate SQL too. When the data volume goes up, we need to pay attention to how DevForce is doing its job.

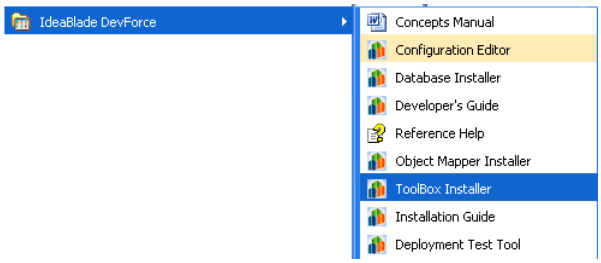
Fortunately there are many tuning options so you can cool down those performance hot-spots. We cover this topic in the Developers Guide. If you need additional help, please contact Customer Support.

Using the Toolbox Installer

There are a number of reasons to install or re-install the DevForce UI visual design components as a separate step:

- The main installer did not add the tab or its items.
- You choose custom installation and omitted support for a tool suite.
- You just purchased a Third-Party control suite and want to add DevForce support.
- You deleted the tab or some of its items and want to re-install them.

The DevForce "Toolbox Installer" is your first option. You will find it in the "IdeaBlade DevForce" folder of your Windows Start menu.

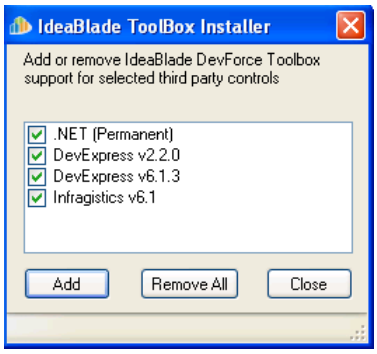


Remember to exit Visual Studio before running any installer.

⇨ Launch the "Toolbox Installer" from the Windows Start Menu.

Dialog presents the control suites available to you. Your edition of DevForce governs your choices. You must have installed the applicable versions of the vendor control suites before you can add DevForce support.

You add or remove support for entire control suites by checking the suite(s) of interest. In this example, we're adding DevForce support for the .NET and Developer Express control suites.



You can add but you cannot remove support for the .NET control suite with this tool.

We typically want to add support with this tool:

⇨ Select the control suite(s)
⇨ Click the [Add] button.

Look closely at the "Status Bar" on the bottom of the form.

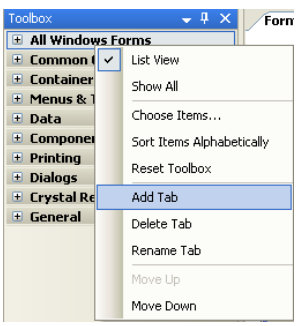
While the tool is working it says "Adding".

When it completes it will say "Added."

⇨ Close the tool when it says "Added".

Manually Installing Toolbox Controls

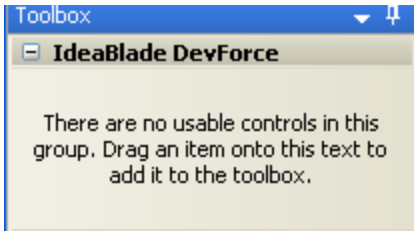
⇨ Launch Visual Studio
⇨ Open the Toolbox in "Design View".
⇨ Right-click any Toolbox tab and select **Add Tab** from the context menu.



Visual Studio creates a new tab with a blank name.

⇨ Name the tab "IdeaBlade DevForce"

Note that the IdeaBlade DevForce tab has no usable controls.



⇨ Right click the tab or tab header and select "Choose Items…" from the context menu

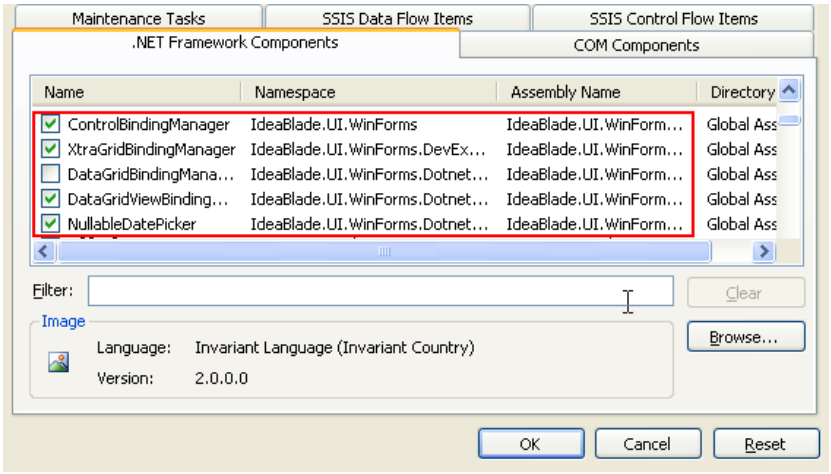
Eventually the "Choose Toolbox Items" dialog appears.

⇨ Select the ".Net Framework Components" tab if not already selected.

⇨ Click the "Namespace" column header to sort by Namespace.

⇨ Scroll to the IdeaBlade assemblies.

⇨ Check the checkboxes next to the components you want to add to the Toolbox.



⇨ Press [OK]

Arrange the items within the tab.

⇨ Drag the DevForce components into your favored positions on the IdeaBlade DevForce Tab.

If you installed them onto the wrong tab …

⇨ Drag the DevForce components from that wrong tab to the IdeaBlade DevForce Tab.