

# Microsoft® Visual Studio® Express Edition 2005 Beta 2 Known Issues

## All Express

**Issue:** Windows Forms SplitContainer Control generates a stack overflow when you set the TabStop property to false.

**Details:** A Windows Forms application that contains a SplitContainer with the TabStop property set to false generates a stack overflow when the Form is displayed. When displayed, the Form briefly appears on the screen and then disappears.

**Workaround:** Set the TabStop property on SplitContainer to true.

**Issue:** Selecting an invalid .NET Framework assembly from the “Choose Toolbox Items” dialog crashes Visual Studio 2005.

**Details:** If an invalid assembly is chosen from when invoking the “Browse...” button on the “.NET Framework Components” tab of the “Choose Toolbox Items” dialog, Visual Studio 2005 will shut down without notice.

**Workaround:** Do not choose an invalid assembly when using this dialog.

**Issue:** Visual Studio 2005 crashes when you attempt to add an existing project to a non-existent solution.

**Details:** If you launch Visual Studio and open a “file” (not a project or solution), and then attempt to add an existing project to the solution, Visual Studio will crash.

**Workaround:** Ensure that you adding a project to a solution that exists, or has been explicitly created when you created a project.

**Issue:** Publish operation fails when invoked using the “Publish Now” or “Publish Wizard” buttons on the Publish tab of the project properties designer.

**Details:** After invoking publishing using either the “Publish Now” or “Publish Wizard” buttons, the publish operation will fail. The error can vary, for example: “Failed to copy file ‘C:\.\setup.exe’ to ‘http://./setup.exe’ . Unable to add ‘setup.exe’ to the Web. Unable to add file ‘setup.exe’ . The system cannot find the file specified” or “Publish failed with the following error: Could not find a part of the path ‘C:\.\publish\.’”

**Workaround:** 1. Close and reopen the project that is to be published.

2. Right-click on the project in the solution explorer, and select ‘Properties.’

3. In project designer, navigate to the ‘Publish tab’. Set properties there as appropriate.

4. Right-click on the project in the solution explorer, and select ‘Publish.’

**Note:** In general, avoid using the “Publish Now” or “Publish Wizard” buttons altogether. Use the ‘Publish’ command surfaced in the context menu in the solution explorer or the ‘Build’ top-level menu item.

**Issue:** Changing an application from full-trust to partial trust causes application to fail at execution after publishing

**Details:** When a previously published application is changed from full-trust to partial-trust and published again, the application fails with a runtime error at execution. The

error reads: "Error: An unexpected error occurred: Failed to grant permission to execute."

**Workaround:** 1. Right-click on the project in the solution explorer and select 'Properties'

2. Select the Security tab on the project properties pane

3. Toggle a property on the page (eg: flip the security zone from "Local Intranet" to "(Custom)" and back)

4. Publish again

**Issue:** Calculating permissions on a Visual Basic or Visual J# partial trust applications from the Security project designer page causes superfluous permissions to be requested.

**Details:** When run on a Visual Basic partial trust application, the permissions calculator on the Security project designer page may return unnecessary permissions. In particular, the list below:

DNSPermission

SocketPermission

KeyContainerPermission

UIPermission

SecurityPermission

ReflectionPermission

EnvironmentPermission

When run on a Visual J# partial trust application, the permissions calculator on the Security project designer page may return superfluous permissions from the list below:

ReflectionPermission

SecurityPermission

UIPermission

**Workaround:** The application code should be inspected and tested to verify whether these permissions returned by the permissions calculator are actually required by the application.

**Issue:** Importing into a project a key file with the same name as a key file already in the project causes Visual Studio to crash

**Details:** From the signing page in the project designer, importing a key from a file when the selected key file has the same name as a key file already in the root directory of the project, will cause Visual Studio to crash.

**Workaround:** Prior to importing a key file into a project, ensure that it has a filename that is distinct from the filenames of other key files already in the root directory of the project.

**Issue:** Upgrading a project that has a user defined class called Settings may causes build errors when building.

**Details:** The generated class for the Settings Designer is not in its own namespace. The compiler thinks MyCustomClass.Settings class is WindowsApplication1.Settings.

**Workaround:** Use a fully qualified name to access user-defined settings class.

**Issue:** Build Error "Property or indexer '%d' cannot be assigned to -- it is read only" when a exiting Visual Studio

**Details:** When settings are edited in the Settings Designer, the auto-generated code is updated. If the user does not save their changes the generated code (Settings.Designer.vb) will be out sync. This is also reproducible with the Resource Editor.

**Workaround:** Run the single file generator on the .settings or .resx file by right-clicking on it in the solution explorer and select Run custom tool

Issue Title:

Adding an Application Configuration File (app.config) to a project causes duplicate messages to be written to the application's File Log

Problem Description:

If you add an Application Configuration File (app.config) to your project, you will see two (seemingly identical) messages written to the application's File Log each time you call `My.Application.Log.WriteEntry` or `WriteException`. The issue is that both the 'baked in' (default) `FileLogTraceListener` instance and the one configured in `app.config` are being added to the default `TraceSource`. The intended behavior is for the instance configured in `app.config` to override (replace) the 'baked in' instance.

Work Around:

There are a couple of options:

Remove the 'baked in' instance from the default `TraceSource` before calling `WriteEntry` or `WriteException`:

```
My.Application.Log.TraceSource.Listeners.Remove("FileLogWriter")
```

Configure the 'baked in' instance programmatically rather than using `app.config` (the following configures the `AutoFlush` property):

```
Dim myFileLog As Logging.FileLogTraceListener = Nothing
Dim i As Integer = 0
With My.Application.Log.TraceSource.Listeners
    While i < .Count AndAlso .Item(i).Name <> "FileLogWriter"
        i += 1
    End While
    If i < .Count Then
        myFileLog = .Item(i)
    End If
End With

myFileLog.AutoFlush = True
```

**Issue:** Adding a Web Reference to a Web Service which returns a data table with the

same name as one already in the project causes a build failure.

**Details:** Attempting to do a build after adding a reference to a Web Service which returns a data table with the same name as one already in the project will fail without reporting the failure in the error list.

**Workaround:** Bring up the project properties and click on the Compile tab. Next click on the "Advanced Compile Options" button and set "Generate serialization assemblies" to "Off" (the default is "Auto").

**Issue:** The auto-generated setup.exe for Setup projects will fail to run the Windows Installer packages when installed from the web.

**Details:** When the auto-generated setup.exe for Setup projects run from the web, it will fail to execute the Windows Installer package after installing the application prerequisites.

If the package is signed, the error message will resemble: "Setup has detected that the publisher of file 'D:\~\Tempoary Internet Files\~\SetupMSI[1].msi' could not be verified."

If the package is not signed, the error message will resemble: "Setup has detected that the publisher of file 'D:\~\Tempoary Internet Files\~\SetupMSI[1].msi' cannot be verified."

**Workaround:** There is no way to avoid this failure if the setup.exe is run from the web; however, one can directly execute the Windows Installer package if the setup.exe fails to run it.

**Issue:** The auto-generated setup.exe fails to install the ClickOnce application when executed from a UNC path.

**Details:** The auto-generated setup.exe will fail to install the ClickOnce application when executed from the UNC share it was published to. Executing the setup.exe directly from the UNC will fail when it attempts to launch the ClickOnce application, with the error message that the application is not formatted correctly.

**Workaround:** If the setup.exe and ClickOnce application is published to a UNC share, have users execute the setup.exe from a hyperlink embedded on a webpage. This link should point to the local relative path of the setup.exe on the UNC share, eg:  
<HREF="setup.exe">Install</A>.

If the install machine is already in the state where it is exhibiting this problem, you will need to shutdown the dfsvc.exe or reboot the machine.

**Issue:** When run from the web, auto-generated setup.exe fails to install an application prerequisite whose installer is a Windows Installer package.

**Details:** When the auto-generated setup.exe is executed from the web, it will fail after downloading a Windows Installer package for an application prerequisite. The error message returned will indicate that setup could not verify the Windows Installer package.

**Workaround:** Install Windows Installer version 2.0 or greater on the machine upon which installation will occur, then re-run the setup.exe on that machine.

**Issue:** ApplicationSettingsBase.Upgrade() throws an exception.

**Details:** If the new ApplicationSettingsBase is used to create settings wrapper classes for an application, the Upgrade() method on ApplicationSettingsBase can be used to bring forward settings from a past version of the application. However, in Beta 2, Upgrade() throws an exception instead of bringing the settings forward.

Note that Upgrade() is automatically called when the first time a setting is accessed on a settings class used within a ClickOnce application that is being run for the first time since the application was upgraded.

**Workaround:** The only work around is to essentially disable Upgrade(). One method, open to any application (non-ClickOnce) that relies on user code calling Upgrade(), is to simply catch and ignore the exception raised. Another method, which can be used by both ClickOnce and non-ClickOnce apps, is to override Upgrade() on the settings wrapper class, without calling base.Upgrade(). Both techniques, of course, will result in no upgrade being performed – the settings values will retain their new version defaults.

**Issue:** Security page in project designer fails after WebBrowser permission excluded

**Details:** For a project open in the IDE:

1. Open the project designer.
2. Select the Security tab.
3. Check “Enable ClickOnce Security Settings”
4. Select the radio button labelled “This is a partial trust application”.
5. Find the WebBrowserPermission in the grid, and select “Exclude” in the Setting column.
6. Close and then re-open the project designer.
7. Select the Security tab.
8. An error message appears on the tab that says “An error occurred trying to load the page. Bad Xml securityElement”

**Workaround:** Take the following steps:

1. Save your changes and close the project.
2. Open the .proj file associated with the project experiencing the problem.
3. Search for and remove the line:  
<ExcludedPermissions>System.Windows.Forms.WebBrowserPermission, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089</ExcludedPermissions>
4. Re-open the project

**Issue:** Auto-generated setup.exe fails when running an application installer located on a UNC fileshare

**Details:** This problem occurs when the auto-generated setup.exe is configured to install the application from a UNC file share (eg: \\machinename\foo). The setup.exe will fail with no error message when attempting to run the application’s installer.

**Workaround:** Take the following steps before re-running the installer:

1. Open a command window on the machine where the setup.exe was published.
2. Navigate to the folder where the setup.exe was published
3. Run the following command to view the current install location:

>setup.exe -url

4. Run the following command to change the install location to file://\

\machinename\folder

>setup.exe -url=file://\machinename\folder

**Issue:** References to 32-bit COM components may not work in VB and C# Applications running on 64-bit platforms.

**Details:** Most existing COM components are only available for 32-bit platforms and will not run in a 64-bit process on a 64-bit platform (although they will run correctly in a 32-bit process on a 64-bit platform). VB and C# applications that reference these 32bit COM components will not run by default on a 64-bit platform because by default the application will launch as a 64-bit process.

The problem appears when a project with one or more COM references is:

1. Migrated to VS2005 and executed on 64 bit platforms or
2. Created using VS2005 on 64 bit platforms

In VS2005 the VB and C# compilers use the platform target property to determine if the.exe or .dll should run in 32 bit or 64 bit CPU architecture mode. The default setting for this property in VS2005 is set to 'AnyCPU' which indicates that the application can run in either 32 and 64 bit mode, depending on the host platform. In this situation you may see an exception such as "Class is not registered ..." when you debug or run these applications.

**Workaround:** The VB and C# Express products do not expose the Target property inside the development environment. You will need to carefully modify the project file using a text or XML editor.

1. Close the project and/or solution
2. Select Open File from the File menu
3. Navigate to the project directory, and highlight the project file
4. Press the Open button, the project file should open in the XML editor
5. Locate the first <PropertyGroup> section and add the following line:  
<PlatformTarget>x86</PlatformTarget>
1. Save the project file
2. Reopen the project and/or solution using Open Project/Solution from the File menu
3. Continue with development, debugging, and testing

Alternatively you can ensure that the COM controls added to the application have 64 bit equivalents on the development and deployment machines, if the application is targeted to 64 bit platforms.

## **Visual C# Express**

**Issue:** Visual Studio may generate non fatal error reports when using Visual C# projects

**Details:** When Visual Studio 2005 encounters an error while processing a Visual C# command, it will generate a non fatal error report. When this happens a window titled “Microsoft Visual C# 2005 IntelliSense” will be shown to the user asking to send the report by clicking on the “Send” button.

When this dialog is shown no data will be lost and the user will be able to continue working in the current session.

The following is an example of a product issue that will generate a non fatal error report:

1. Create a C# ASP.Net web site.
2. Change the web page view to design mode.
3. Drag a button from the toolbar to the web page.
4. Double click the button, this will open the code beside file and generate an event handler.
5. In the editor navigation bar member dropdown select Button1.

**Workaround:** The user will be able to continue working once the error dialog has been dismissed.

**Issue:** The Visual C# personal snippets directory will be renamed in the final release of Visual Studio 2005

**Details:** In the final release of Visual Studio 2005 the personal snippets directory will be renamed from “VC#” to “Visual C#”. The personal snippets directory is located under “My Documents\Visual Studio 2005\Code Snippets”.

If the user has created snippets in Beta 2, uninstalls Beta 2, then installs the final release of Visual Studio 2005, there will be two directories at the above mentioned location (“Visual C#” and “C#”). The user will need to copy the snippets from the “VC#” directory to the “Visual C#” directory.

**Workaround:** After installing the final release of Visual Studio 2005, copy any user-defined snippets from the “VC#” directory to the “Visual C#” directory, then delete the “VC#” directory

**Issue:** Using the C# “Switch” snippet can hang Visual Studio

**Details:** When a user invokes the Visual C# “Switch” expansion, types the name of an enum variable or field in the editable field and presses the Enter key, Visual Studio will generate a case statement for each member of the enum.

In certain cases Visual Studio may hang when doing this.

## **Visual Basic Express**

**Issue:** Community Content Installer doesn’t allow installation of Visual Basic Code Snippets

**Details:** User attempts to install a Code Snippet by clicking on a .vsi file containing Visual Basic Code Snippets. When the user is asked to select a location to install the Code Snippets, they see the message “No registered snippet locations”.

**Workaround:** Rename the .vsi file to .zip and extract the file to a folder on disk.

In Visual Studio, go to Tools.Code Snippets Manager and click the Import... button.

Choose the code snippets that were extracted and click open.

Choose an installation folder for each of the Code Snippets and click the Finish button.

**Issue:** Application Log file VBEvents.Log is empty for projects upgraded from Visual Basic 6.0

**Details:** When a Visual Basic 6.0 Project That writes to the VBEvents.log is upgraded to use My.Application.Log.WriteEntry the log file will be empty.

**Workaround:** Enable the Application Framework by opening the project properties via the Project Menu and on the Application tab check the “Enable application framework” check box.

**Issue:** My.Application.Log is not working ‘out of the box’ for Console Applications

**Details:** Please note that this issue only applies to Console Applications.

If you write messages containing less than 1024 bytes of data to a File Log using My.Application.Log.WriteEntry or WriteException, some (or all) of these messages will not be written to the application’s File Log.

**Workaround:** You will need to call My.Application.Log.TraceSource.Flush() before you application terminates.

**Issue:** On IA64 and AMD64 WS03 SP1, if an unhandled exception occurs in a Windows Application’s Form Load event handler, we will not break into the debugger nor will the Exception Assistant UI appear.

**Details:** If there is an unhandled exception that is thrown in the Form Load Event Handler, an Exception message will get printed to the Immediate Window (or Output window if you do not have your output redirected to the Immediate Window). However, your application will not break into the debugger nor will the Exception Assistant UI appear to give you a chance to fix your code.

**Workaround:** You should add structured Exception Handling to your code in the Form Load event handler. Any exceptions that occur will get caught and you can debug your application (if you had set BP’s) in your Catch/Finally block.

## **Visual Web Developer Express**

**Issue:** Unable to resize certain web controls in the designer surface

**Details:** When trying to resize a web control (hyperlink, label, check box, link button, and radio button) the designer shows a watermark of the new size as you resize the control, but the control never shows it’s resize. The controls persist the new size in code, but don’t show the size in design time or run time.

**Workaround:** Remove the DOCTYPE line from the top of the html page, controls will then render at their expected size.

**Issue:** Failed to update database "...\\App\_Data\\Database.mdf" because the database is read-only.

**Details:** When attempting to update data in a SQL Server Express database, it is possible to encounter this exception when the database has been attached as a read-only database. When an MDF database file is attached to SQL Server Express, the permissions granted to the database will be based on the file ACLs of the process identity that does the attach. If the ACLs on the MDF grant only Read permission to the process identity but not Write



permission, the database will be attached to SQL Server Express as a read-only database. You will not be able to change these permissions until you first detach the database from the attached SQL Server Express instance.

**NOTE:** ASP.NET runs as the Network Service account on IIS 6.0 and as a local ASPNET account on IIS 5.1. In order to support updating a SQL Server Express database, this process account must be granted both Read and Write permission to the database prior to attach.

**Workaround:** In order to change the ACLs on the MDF after it has been attached, for example to grant Write permission to the ASP.NET process identity, you need to first detach the database from the attached SQL Server Express instance. Follow the steps below, assuming an ASP.NET application running under the Network Service identity:

1. Download SSEUtil.exe from the following URL:

<http://go.microsoft.com/fwlink/?LinkId=44178>

2. From a command prompt, run the following statement:

```
> SSEUtil.exe -child "NT AUTHORITY\NETWORK SERVICE" -detach "<full path to database file>"
```

Sample output:

Instance '\\.\pipe\FBBD883D-4280-42\tsql\query' was found.

Detached 'd:\Inetpub\wwwroot\...\App\_Data\Database2.mdf' successfully.

If SSEUtil.exe fails to detach the database, it may be because the SQL Server Express instance has timed out and is not running. Try requesting the ASP.NET page that attaches to the database again and then retry the detach command.

After successfully running the above command you may change the ACLs on the MDF file (and associated LDF file, if one exists) and reattach the database by running the application again:

1. Right-click the MDF files and choose Properties...
2. Select the Security tab
3. Click the Add... button
4. Click the Locations... button and choose the local machine name (top of list) and click OK
5. Enter "Network Service" under object names and click OK
6. Back in Properties dialog, Select NETWORK SERVICE
7. Click Allow checkbox next to Write permission and click OK
8. Repeat 1-7 for the associated LDF file

**Issue:** RESX files opened using the managed RESX editor in a Web Application generate a warning that can be ignored. Double clicking on the message may result in an

"Object Reference Not Set To An Instance" error message

**Details:** RESX files in a Web Application have a warning listed in the Error List, whenever the user opens the RESX file in the managed RESX editor. This editor is the default editor for this file type in Visual Studio. There are several ways to create RESX files within a web application:

- Using Generate Local Resources, when viewing a web page, master page, content page or user control in design view.
- Using The "Add New Item" command from the solution and selecting "Assembly Resource File"

The warning "Double-click here to enable strongly-typed resources for this file" is added to the Error List. If the developer does not follow the instructions in the warning message, or immediately closes the managed RESX editor, then the warning will persist. If the developer double-clicks on the warning immediately, then the warning will be removed.

When not removed, this warning message is persistent and cumulative:

- Opening and closing the same RESX will simply add more copies of the warning to the Error List
- Deleting the RESX file will not clear the warning

Developers can ignore this warning message immediately to clear the problem. Double clicking on the persistent warning may result in an "Object Reference Not Set To An Instance" error message.

Developers should simply clear the warning by following the instructions in the warning immediately. Following the warning instructions will not perform any operation other than clearing the warning.

**Workaround:** When the managed RESX editor is opened on a RESX file, immediately clear the warning message by double-clicking the warning.

If persistent warnings are listed in the "Error List", then the "Error List" can be cleared by closing the IDE and re-opening the project, or these warnings simply ignored.

**Issue:** Visual Studio may generate non fatal error reports when using Visual C# projects

**Details:** When Visual Studio 2005 encounters an error while processing a Visual C# command, it will generate a non fatal error report. When this happens a window titled "Microsoft Visual C# 2005 IntelliSense" will be shown to the user asking to send the report by clicking on the "Send" button.

When this dialog is shown no data will be lost and the user will be able to continue working in the current session.

The following is an example of a product issue that will generate a non fatal error report:

1. Create a C# ASP.Net web site.
2. Change the web page view to design mode.
3. Drag a button from the toolbar to the web page.
4. Double click the button, this will open the code beside file and generate an event

handler.

5. In the editor navigation bar member dropdown select Button1.

**Workaround:** The user will be able to continue working once the error dialog has been dismissed.

**Issue:** The Visual C# personal snippets directory will be renamed in the final release of Visual Studio 2005

**Details:** In the final release of Visual Studio 2005 the personal snippets directory will be renamed from “VC#” to “Visual C#”. The personal snippets directory is located under “My Documents\Visual Studio 2005\Code Snippets”.

If the user has created snippets in Beta 2, uninstalls Beta 2, then installs the final release of Visual Studio 2005, there will be two directories at the above mentioned location (“Visual C#” and “C#”). The user will need to copy the snippets from the “VC#” directory to the “Visual C#” directory.

**Workaround:** After installing the final release of Visual Studio 2005, copy any user-defined snippets from the “VC#” directory to the “Visual C#” directory, then delete the “VC#” directory

**Issue:** IDE crash when accessing chrome after converting Login control to template with an incorrect theme file

**Details:** If a user's web application contains a .skin file with a malformed LayoutTemplate property defined for the Login control, then the Visual Web Developer environment may crash if the user converts a Login control in one of the application's pages to template format, and subsequently manipulates the control in the design view.

**Workaround:** To avoid this issue, it is sufficient that users ensure their .skin files are well-formed and have opening and closing tags for all inner properties

**Issue:** Using the C# “Switch” snippet can hang Visual Studio

**Details:** When a user invokes the Visual C# “Switch” expansion, types the name of an enum variable or field in the editable field and presses the Enter key, Visual Studio will generate a case statement for each member of the enum.

In certain cases Visual Studio may hang when doing this.

## **Visual C++ Express**

**Issue:** ASP .NET Web Service Project Does Not Compile

**Details:** When creating a new ASP .NET Web Service Project, the resulting code generated will not compile, build or run. This project template is not supported in Visual C++ Express and will be removed for the final release.

**Issue:** Cannot make data access calls from SQL/CLR C++/CLI projects.

**Details:** The customers will see a compiler error when they try to create an instance of System::Data::SqlClient::SqlConnection class from system.data.dll assembly. The compiler error will be:

t.cpp(10) : error C3624: 'System::EnterpriseServices::ITransaction': use of this type requires a reference to assembly 'System.EnterpriseServices'

This error is caused by the presence of the following function in SqlConnection class:  
.method public hidebysig virtual instance void EnlistDistributedTransaction(class [System.EnterpriseServices]System.EnterpriseServices.ITransaction transaction)

Since SQL Server does not have System.EnterpriseServices.dll in its .NET Runtime, user's project cannot add a reference to this dll to get around the compiler error.

**Workaround:** There are 2 workarounds to this issue:

1. Users need to add the following 3 lines of code into their SQL/CLR project where they're making an instance of SqlConnection class and recompile.

```
namespace System
{
    namespace EnterpriseServices
    {
        public interface class ITransaction
        {
        };
    }
}
#pragma warning( push )
#pragma warning( disable : 4691 )
    SqlConnection^ connect = gcnew SqlConnection();
#pragma warning( pop )
```

2. Users can deploy their local .NET Runtime copy of System.EnterpriseServices.dll on to SQL Server with their project and add a reference to that DLL in the project.

**Issue:** Cannot set VC++ Directories in the Options dialog of Visual C++ Express Edition

**Details:** The VC++ Directories page on the Options dialog does not have a list box for viewing and modifying the directories used during a build. Consequently, it is not possible to modify the build directories using the IDE UI.

**Workaround:** C++ Directories can be added, removed or changed by modifying the VCProjectEngine.dll.express.config file located in the \vc\vcpackages subdirectory of the Visual C++ Express Edition install location in a text editor, saving the changes and restarting the Visual C++ Express Edition.

## **Visual Studio Integrated Development Environment (IDE)**

**Issue:** Error message when viewing exported template files with non-ASCII characters in the filename

**Details:** If a user creates a template which has Unicode characters in its file name (e.g.

ÀĀĒĒ), they will be unable to open the exported template (ÀĀĒĒ.zip) in explorer. Attempting to do so will generate an error message: “The compressed (zipped) folder is invalid or corrupted.”

This is caused by the design of the ZIP application which does not allow non-ASCII characters in file names.

**Workaround:**

The user will need to rename the zip file to have an ASCII only file name and path to open the zip file directly.

Note: This workaround is only required for opening the template in explorer, the non-ASCII name will work correctly in Visual Studio.

## **.NET Framework**

**Issue:** Unicode Surrogate Behavior with ASP.NET Providers that use SQL Server

**Details:** ASP.NET providers that use SQL Server are constrained to the level of Unicode surrogate support provided by SQL Server 7.0 and SQL Server 2000. SQL Server stores and retrieves surrogate pairs losslessly. However, there is no linguistic comparison for surrogates. Surrogate characters are ignored during comparison operations and during uniqueness checks.

**Issue:** AuthorizationStoreRoleProvider: Default value for ApplicationName results in an error from Authorization Manager.

**Details:** Authorization Manager does not allow the "/" character in application names. In the absence of applicationName being set in configuration, the AuthorizationStoreRoleProvider follows the same logic used by other ASP.NET providers to determine a default value for applicationName. When running within ASP.NET, this results in a value that starts with the "/" character.

**Work Around:** Always set the applicationName attribute in configuration when using the AuthorizationStoreRoleProvider within an ASP.NET application.

**Issue:** AuthorizationStoreRoleProvider: Incorrect or obscure errors are returned from Authorization Manager

**Details:** The AuthorizationStoreRoleProvider depends on Authorization Manager. Not all error messages returned from AuthorizationManager indicate the root cause of a problem. Listed below are known error messages that are either incorrect or vague.

[COMException (0x8007052b): Unable to update the password. The value provided as the current password is incorrect.]

This error is really an access denied error. If all of the following conditions are met, this error can occur: ASP.NET is deployed on IIS 5.0, Windows XP IIS 5.1, or in IIS 5.0 isolation mode on Windows Server 2003, the application is configured to use Integrated Windows Authentication, and the policy file is located inside of the directory structure of the current ASP.NET application. If ASP.NET is running as a local machine account and attempts to access a policy store in a remote AD or ADAM instance this error can also occur.

[More data is available]

This error really means that the policy store could not be found. If the connection string points at an ADAM instance, but the connection string references an ADAM partition that does not exist, this error can occur. For example, in the following connection string “LDAP://localhost:4000/Cn=storename, DC=Partition1”, if Partition1 does not exist in the ADAM instance, this error will occur.

[The specified server cannot perform the requested operation]

This error really means that the specified server could not be found. If the connection string points at a non-existent server, or uses a port number that AD or ADAM are not listening on, this error can occur.

[ArgumentException: The parameter is incorrect]

This error message really indicates that an LDAP query group in Authorization Manager used to determine if a user is in a role is invalid.

**Work Around:** For each error condition listed above, review the possible causes. If an application’s configuration matches one of the possible causes, change or fixes the application’s configuration based upon the information listed above.

**Issue:** ActiveDirectoryMembershipProvider cannot securely connect to ADAM over SSL when running on either Windows 2000 SP4 (or lower) or Windows XP SP1 (or lower)

**Details:** The ActiveDirectoryMembershipProvider internally relies on ADSI. On older versions of both Windows 2000 and Windows XP, ADSI does not properly handle SSL connections to an ADAM instance. As a result the ActiveDirectoryMembershipProvider will fail initialization when attempting a secure connection to ADAM using SSL.

**Work Around:** On Windows XP, the customer should upgrade to SP2.

On Windows 2000 the customer should either install SP5 when it becomes available, or should obtain the QFE described in KB 817583.

**Issue:** ASP.NET Membership Enforces Password Strength Requirements

**Details:** Prior to Beta 2, it was possible to create weak passwords in the default configuration of the ASP.NET Membership feature. In Beta 2, basic password strength enforcement has been implemented for Membership. The default configuration requires all new passwords to have a minimum length of seven characters, and at least one character must be a non-alphanumeric character ( [ ] ! @ # \$ etc...). Pre-existing passwords will continue to work unchanged though.

**Work Around:** The password strength configuration can be configured for both the SQL and Active Directory Membership providers by changing the values of the following three new provider attributes in configuration:

minRequiredPasswordLength (defaults to seven)  
minRequiredNonalphanumericCharacters (defaults to one)  
passwordStrengthRegularExpression (no default value in configuration)

**Issue:** Sql Server Express User Instances Should be Disabled on Shared Hosting Machines

**Details:** ASP.NET 2.0 integrates with SQL Server Express 2005 to provide automatic creation of the database required by many of ASP.NET 2.0's new application services. This functionality relies on SQL Server Express' support for spawning server processes that run with the identity of either the interactive user or the identity of the worker process hosting ASP.NET. In un-trusted environments such as on a shared hosting server, the ability to spawn SQL Server worker processes should not be enabled due to the potential for unintentionally sharing data between ASP.NET applications .

**Work Around:** The user instancing functionality in SQL Server Express 2005 can be disabled with the following steps:

1. While logged in as a local box administrator, open up a command window by running cmd
2. If osql.exe is not available from any directories listed in the PATH environment variable, change directories to the SQL Server Express directory that contains osql.exe.
3. Connect to the parent instance of SQL Server: osql -E -S .\sqlexpress
4. Issue the following SQL commands:  
exec sp\_configure 'show advanced option', '1'  
go  
reconfigure with override  
go  
exec sp\_configure 'user instances enabled', 0  
go  
reconfigure with override  
go

**Issue:** Using OracleParameterCollection.CopyTo() with an Array of OracleParameter objects results in a StackOverflow.

**Details:** Any attempt to copy the OracleParameterCollection to an array of OracleParameter objects will result in a StackOverflow. This is a regression from Everett. The workaround is for a customer to perform the cast to System.Array themselves.

When using the CopyTo() method, if you do not cast to a System.Array, you will see a StackOverflow. This is due to a bug in the CopyTo() method.

**Workaround:** Cast the OracleParameter Array to a System.Array when calling the CopyTo method. This will stop the StackOverflow from occurring.

**Issue:** Intra Whidbey Obsolete APIs will be removed by V2.0 release

**Details:** In the Beta2 version of the .NET Framework product, there are various APIs that have been added in since V1.1, but are marked obsolete. All of these APIs will be removed from the product at V2.0 release. These are APIs which were added during the beta cycles of the V2.0 product lifetime, but as a result of design changes and/or product feedback, have subsequently been replaced, or will simply be removed.

It is paramount for your ability to easily move forward to the final V2.0 release of the product that these obsolete APIs not be referenced.

Any APIs which were either a) already obsolete in V1.x, or b) existed in V1.1 and were marked as obsolete in V2.0, will not be removed. This action will only affect APIs that are marked obsolete, and have been introduced as new APIs in V2.0.

See <http://go.microsoft.com/fwlink/?LinkId=43753> for the list of relevant APIs.

**Workaround:** Do not use any of the referenced obsolete APIs. An efficient way of ensuring you achieve this is to be warning clean for obsolescence (that is, you have no warnings regarding usage of obsolete APIs).

**Issue:** Updates on the .NET Profiling API

**Details:** Late coming updates on using the .NET Profiling API

**Workaround:** Look on

<http://lab.msdn.microsoft.com/vs2005/teams/clr/profilebeta2.aspx> for updates on this API.

**Issue:** Windows Forms SplitContainer Control generates a stack overflow when you set the TabStop property to false.

**Details:** A Windows Forms application that contains a SplitContainer with the TabStop property set to false generates a stack overflow when the Form is displayed. When displayed, the Form briefly appears on the screen and then disappears.

**Workaround:** Set the TabStop property on SplitContainer to true.

**Issue:** Selecting an invalid .NET Framework assembly from the “Choose Toolbox Items” dialog crashes Visual Studio 2005.

**Details:** If an invalid assembly is chosen from when invoking the “Browse...” button on the “.NET Framework Components” tab of the “Choose Toolbox Items” dialog, Visual Studio 2005 will shut down without notice.

**Workaround:** Do not choose an invalid assembly when using this dialog.

**Issue:** Automatic creation of provider databases with SQL Server Express 2005 user instancing is only available for ASPNET, NETWORK SERVICE and INTERACTIVE

**Details:** The ASP.NET automatic MDF creation process for SQL Server providers only works when IIS is running either as the local ASPNET machine account, or as NT AUTHORITY\NETWORK SERVICE. The automatic database creation process also works when a developer is interactively developing using Cassini. However, when developing against IIS using either a different local machine account, or a domain user account, the automatic database creation process will fail.



**Workaround:** There is no workaround for using arbitrary domain or local machine accounts in Beta 2.

**Issue:** Assemblies with security-transparent code that are NGen'd and run on 64-bit CLR will crash.

**Problem Description:** If you have an assembly that uses security-transparent code (a new feature in .NET Framework 2.0), and you NGen that assembly and run it on a 64-bit version of the CLR, that assembly will cause an access violation.

**Work Around:** Do not NGen any assemblies that use security-transparent code and run on 64-bit versions of the CLR.

## **Crystal Reports**

For issues with Crystal Reports for Visual Studio 2005, please see the Readme at:  
[http://www.businessobjects.com/products/dev\\_zone/net/VisualStudio2005Beta](http://www.businessobjects.com/products/dev_zone/net/VisualStudio2005Beta)