

Monday, March 24. 2014

Be careful using ThreadStatic in Visual Studio unit tests

Given is a ThreadStatic field which is set to empty in the static constructor, and set by a public instance method:

```
public class TestClass
{
    [ThreadStatic]
    private static string threadStaticValue;
    static TestClass()
    {
        threadStaticValue = "";
    }
    public void SetThreadStaticValue(string value)
    {
        threadStaticValue = value;
    }
    public string GetThreadStaticValue()
    {
        return threadStaticValue;
    }
}
```

Now two unit tests check whether the threadStaticValue is set correct.

```
[TestClass]
public class ThreadStaticTest
{
    [TestMethod]
    public void WithThreadStaticValueSetTest()
    {
        Debug.WriteLine("Thread id: {0}", System.Threading.Thread.CurrentThread.ManagedThreadId);
        TestClass tc = new TestClass();
        tc.SetThreadStaticValue("foo");
        Assert.AreSame(tc.GetThreadStaticValue(), "foo");
    }
    [TestMethod]
    public void WithThreadStaticValueNotSetTest()
    {
        Debug.WriteLine("Thread id: {0}", System.Threading.Thread.CurrentThread.ManagedThreadId);
        TestClass tc = new TestClass();
        Assert.AreSame(tc.GetThreadStaticValue(), "");
    }
}
```

Those tests will fail(good) or pass(bad) depending of whether the test engine will reuse the same thread. Executed a second time, the result could be different.

Solution:

Don't use ThreadStatic at all

Make sure the ThreadStatic fields are initialized correctly before using them

Friday, October 5, 2012

Start VS2012 .sln and .proj files always as Administrator on Windows 8

VS2012 non admin problem description

I'm using windows 8 and visual studio 2012, and i had some strange problems related to permission.
If visual studio runs under non-administrator rights, it cannot do some compile tasks like "Register for COM interop"

Step by step solution

Start a VisualStudio command prompt

Start a VisualStudio command prompt and navigate it to the place where devenv.exe is located. On my System this is
C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE

Create a manifest file from devenv.exe

Execute the following command:

```
mt -inputresource:"devenv.exe" -out:devenv.exe.manifest
```

This will create the file devenv.exe.manifest

Edit the content of devenv.exe.manifest

Change the level value of requestedExecutionLevel from asInvoker to requireAdministrator

Write the manifest back into the devenv.exe

Execute the following command in the visual studio command prompt:

```
mt -outputresource:devenv.exe -manifest devenv.exe.manifest
```

Check whether visual studio 2012 starts as administrator

Just double click any .sln or .proj file, and check whether visual studio starts as administrator or not.

Posted by suntsu in .net at 06:49

Wednesday, June 13, 2012

Howto set MinWidth or MinHeight of a WinRT ScrollViewer

I use the ScrollViewer control in a windows 8 metro app in order to zoom in/out an image. The problem now is, that it is possible to zoom in the image untill you almost can't zoom out again(because the image gets so small). I've tried MinWidth and MinHeight on the ScrollViewer and on the image, but it didn't work. After some time i found the MinZoomFactor of the ScrollViewer, and this now works. It was just not obvious to me...

Posted by suntsu in .net at 19:07

Wednesday, January 25, 2012

Exception: That assembly does not allow partially trusted callers

I have .net 3.5 program where i load some assemblies from a network share via reflection. Now i've got the following exception: "That assembly does not allow partially trusted callers"(Detailed message below).
The calling assembly(.exe) is local. This .exe loads the assembly from network via reflection. The assembly from the network share then has a reference on a third assembly which is located beneath the .exe. (Hard to explain, image will do better...)

The reason for this failure is the .net security. It doesn't allow the MyLib.dll to be loaded, because the calling assembly is located on a network share and is signed with a strong name.

Solution 1, caspol.exe

Allow all code to be executed.

This might be dangerous, but if you're desperate it could be a solution.

```
"C:\Windows\Microsoft.NET\Framework\v2.0.50727\CasPol.exe" -machine -chgggroup All_Code FullTrust
```

Solution 2, no signing

If you remove the signature from MyLib2.dll, everything works.

Solution 3, set attribute AllowPartiallyTrustedCallers

If you set the AllowPartiallyTrustedCallers in the AssemblyInfo.cs of the signed assembly, everything works.

using System.Security;

```
[assembly: AllowPartiallyTrustedCallers()]
```

Usefull links

caspol doku

howto set fulltrust to a share

Using Libraries from Partially Trusted Code

Allowing Partially Trusted Callers

Full exception(for google):

```
System.Reflection.TargetInvocationException: Exception has been thrown by the target of an invocation. --->  
System.IO.FileNotFoundException: Could not load file or assembly 'PermissionCaspolTestLibSigned, Version=1.0.0.0,  
Culture=neutral, PublicKeyToken=null' or one of its dependencies. The system cannot find the file specified.  
File name: 'PermissionCaspolTestLibSigned, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null'  
    at PermissionCaspolTestLib.Class1..ctor()
```

WRN: Assembly binding logging is turned OFF.

To enable assembly bind failure logging, set the registry value [HKLM\Software\Microsoft\Fusion!EnableLog] (DWORD) to 1.

Note: There is some performance penalty associated with assembly bind failure logging.

To turn this feature off, remove the registry value [HKLM\Software\Microsoft\Fusion!EnableLog].

--- End of inner exception stack trace ---

```
at System.RuntimeTypeHandle.CreateInstance(RuntimeType type, Boolean publicOnly, Boolean noCheck, Boolean&
canBeCached, RuntimeMethodHandle& ctor, Boolean& bNeedSecurityCheck)
at System.RuntimeType.CreateInstanceSlow(Boolean publicOnly, Boolean fillCache)
at System.RuntimeType.CreateInstanceImpl(Boolean publicOnly, Boolean skipVisibilityChecks, Boolean fillCache)
at System.Activator.CreateInstance(Type type, Boolean nonPublic)
at PermissionCaspolTest.Program.DoSth()
```

----- Inner Exception -----

```
System.IO.FileNotFoundException: Could not load file or assembly 'PermissionCaspolTestLibSigned, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=null' or one of its dependencies. The system cannot find the file specified.
File name: 'PermissionCaspolTestLibSigned, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null'
at PermissionCaspolTestLib.Class1..ctor()
```

WRN: Assembly binding logging is turned OFF.

To enable assembly bind failure logging, set the registry value [HKLM\Software\Microsoft\Fusion!EnableLog] (DWORD) to 1.

Note: There is some performance penalty associated with assembly bind failure logging.

To turn this feature off, remove the registry value [HKLM\Software\Microsoft\Fusion!EnableLog].

Posted by suntsu in .net at 08:41

Friday, January 20. 2012

Uploading large files to IIS7(asp.net or mvc)

You can upload files with a size of up to 2GB, but it requires some modifications in your web.config.

...

...

Posted by suntsu in .net at 12:03

Wednesday, January 18, 2012

What to do if OnOrientationChanged is not fired on windows phone 7 emulator?

If you debug your windows phone application, and the OnOrientationChanged event is not fired, just press Page Up or Page Down. Those keys are used to activate/deactivate the phone keyboard. If the phone keyboard is active, the following code will be executed. Strange thing...

```
protected override void OnOrientationChanged(OrientationChangedEventArgs e)
{
    base.OnOrientationChanged(e);
}
```

Found(after some time) on the microsoft forum.

Posted by suntsu in .net, MySmug at 18:27

Tuesday, September 6. 2011

Get the current directory from a c# application

Sometimes i have to read/write files from the root directory of my application.

Usually i did this with `Environment.CurrentDirectory`.

Unfortunately, this only works if the application is a standalone application(.exe). In case the application is a windows service or a asp.net website, the path from `Environment.CurrentDirectory` is wrong.

There are several ways to get the root directory of an application, but so far i found only one which works in all scenarios.

The only one which always works is:

//works...

```
IO.Path.GetDirectoryName(new  
System.Uri(System.Reflection.Assembly.GetExecutingAssembly().CodeBase).LocalPath)
```

Here is a (incomplete) list of ways to get the root path of a c# application:

1. Windows service

`Environment.CurrentDirectory:`

```
//C:\Windows\system32
```

`System.IO.Path.GetDirectoryName(new`

```
System.Uri(System.Reflection.Assembly.GetExecutingAssembly().CodeBase).LocalPath)
```

```
//D:\project\WindowsServiceTest\WindowsServiceTest\bin\Debug
```

`AppDomain.CurrentDomain.BaseDirectory:`

```
//D:\project\WindowsServiceTest\WindowsServiceTest\bin\Debug\
```

```
Process.GetCurrentProcess().MainModule.FileName.Substring(0,  
Process.GetCurrentProcess().MainModule.FileName.LastIndexOf("\")):
```

```
//D:\project\WindowsServiceTest\WindowsServiceTest\bin\Debug
```

2. Console application

`Environment.CurrentDirectory:`

```
//D:\project\ConsoleApplication3\ConsoleApplication3\bin\Debug
```

`System.IO.Path.GetDirectoryName(new`

```
System.Uri(System.Reflection.Assembly.GetExecutingAssembly().CodeBase).LocalPath):
```

```
//D:\project\ConsoleApplication3\ConsoleApplication3\bin\Debug
```

`AppDomain.CurrentDomain.BaseDirectory:`

```
//D:\project\ConsoleApplication3\ConsoleApplication3\bin\Debug\
```

```
Process.GetCurrentProcess().MainModule.FileName.Substring(0,
```

```
Process.GetCurrentProcess().MainModule.FileName.LastIndexOf("\")):
```

```
//D:\project\ConsoleApplication3\ConsoleApplication3\bin\Debug
```

3. asp.net mvc3 website vs2010 integrated webserver (Debug mode)

`Environment.CurrentDirectory:`

```
//C:\Program Files (x86)\Common Files\Microsoft Shared\DevServer\10.0
```

`System.IO.Path.GetDirectoryName(new`

```
System.Uri(System.Reflection.Assembly.GetExecutingAssembly().CodeBase).LocalPath):
```

```
//D:\project\MvcApplication1\MvcApplication1\bin
```

`AppDomain.CurrentDomain.BaseDirectory:`

```
//D:\project\MvcApplication1\MvcApplication1\
```



```
Process.GetCurrentProcess().MainModule.FileName.Substring(0,
Process.GetCurrentProcess().MainModule.FileName.LastIndexOf("\"):
//C:\Program Files (x86)\Common Files\Microsoft Shared\DevServer\10.0
```

4. asp.net mvc3 website IIS (Release mode)

```
Environment.CurrentDirectory:
//c:\windows\system32\inetsrv
System.IO.Path.GetDirectoryName(new
System.Uri(System.Reflection.Assembly.GetExecutingAssembly().CodeBase).LocalPath):
//d:\project\MvcApplication1\MvcApplication1\bin
AppDomain.CurrentDomain.BaseDirectory:
//d:\project\MvcApplication1\MvcApplication1\
Process.GetCurrentProcess().MainModule.FileName.Substring(0,
Process.GetCurrentProcess().MainModule.FileName.LastIndexOf("\"):
//c:\windows\system32\inetsrv
```

Posted by suntsu in .net at 11:25

Tuesday, May 10, 2011

c# and vb.net sample to remove eventhandler with reflection

This is a sample how to remove a (non-public) eventhandler in c# and vb.net.

```
//c# sample
using System;
using System.Reflection;
using System.Windows.Forms;

namespace WindowsFormsApplication5
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            Form2 form2 = new Form2();
            //remove eventhandler with reflection
            EventInfo ei = form2.GetType().GetEvent("Load");
            Delegate d = Delegate.CreateDelegate(typeof(EventHandler), form2, "Form2_Load", false);
            ei.RemoveEventHandler(form2, d);

            //messagebox is not shown, since the eventhandler for load was removed
            form2.ShowDialog();
        }
    }
}

using System;
using System.Windows.Forms;
namespace WindowsFormsApplication5
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            this.Load += new EventHandler(Form2_Load);
        }

        void Form2_Load(object sender, EventArgs e)
        {
            MessageBox.Show("Hello World");
        }
    }
}
```

```
'vb.net sample
Imports System.Reflection
```

```
Public Class Form1
    Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
```

```
Dim myForm As New Form2
Dim type As Type = GetType(Form2)

'remove eventhandler with reflection
Dim eventInfo As EventInfo
eventInfo = type.GetEvent("Load")
Dim loadHandler As System.Delegate
loadHandler = System.Delegate.CreateDelegate(GetType(EventHandler), myForm, "Form2_Load", False)
eventInfo.RemoveEventHandler(myForm, loadHandler)

'messagebox is not shown, since the eventhandler for load was removed
myForm.ShowDialog()
```

```
End Sub
End Class
```

```
Public Class Form2
    Private Sub Form2_Load(sender As Object, e As System.EventArgs) Handles Me.Load
        MessageBox.Show("hello world")
    End Sub
End Class
```

RemoveEventHanlderWithReflection.zip

Posted by suntsu in .net at 06:15

Friday, February 25. 2011

Generic c# Json serializer/deserializer sample

```
public class JsonSerializer {  
    /// Serializes a given object into a json string  
    /// Type to serialize  
    /// Object to serialize  
    /// Serialized Json string of the given object  
    public static string SerializeToJson(T objectToSerialize) where T : class {  
        if(objectToSerialize==null) throw new ArgumentException("objectToSerialize must not be null");  
        MemoryStream ms = null;  
        try {  
            DataContractJsonSerializer serializer = new DataContractJsonSerializer(objectToSerialize.GetType());  
            ms = new MemoryStream();  
            serializer.WriteObject(ms, objectToSerialize);  
            return Encoding.UTF8.GetString(ms.ToArray());  
        } finally {  
            if (ms != null) {  
                ms.Close();  
            }  
        }  
    }  
    /// Deserialize a given json string  
    /// Type to deserialize  
    /// Json String representation  
    /// Deserialized object  
    public static T DeserializeFromJson(string jsonStringToDeserialize) where T : class {  
        if(string.IsNullOrEmpty(jsonStringToDeserialize)) throw new ArgumentException("jsonStringToDeserialize must not be null");  
        MemoryStream ms = null;  
        try {  
            ms = new MemoryStream(Encoding.UTF8.GetBytes(jsonStringToDeserialize));  
            DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(T));  
            return (T)serializer.ReadObject(ms);  
        } finally {  
            if (ms != null) {  
                ms.Close();  
            }  
        }  
    }  
}
```

Posted by suntsu in .net at 09:03

Wednesday, February 9, 2011

MSBuild problem: The type or namespace name 'Linq' does not exist in the namespace 'System' (are you missing an assembly reference?)

Problem: I tried to build a .net(c#) solution with MSBuild, and got the following error: The type or namespace name 'Linq' does not exist in the namespace 'System' (are you missing an assembly reference?) Even with a valid using to System.Linq. The problem seems to be Visual Studio. It seems Visual Studio implicit adds a reference to System.Core, whereas MSBuild does not. The screenshot shows a small solution, which should be built by MSBuild. Buildscript used(master.proj):

```
true
  Rebuild
  Debug
```

```
#Batch to start the build
SET MSBUILD="C:\WINDOWS\Microsoft.NET\Framework\v3.5\msbuild.exe"
SET LogDll="c:\windows\Microsoft.Net\Framework\v2.0.50727\Microsoft.Build.Engine.Dll"
%MSBUILD% master.proj /tv:3.5 /target:CompileTest
/!:FileLogger,Microsoft.Build.Engine;logfile=MyLog.log;verbosity=q;encoding=utf-8 /P:NOWARN=
pause
```

Building this, the following error is shown from MSBuild:

Solution:

Add via notepad(++) a reference to System.Core directly to the csproj/vbproj file:

After that, build with MSBuild works fine.

If anyone has a better solution for this problem, please leave a comment.

À

cheers

Posted by suntsu in .net at 13:25

Tuesday, February 23. 2010

Attach process at runtime in visual studio 2008

I have a wcf service which has to be unittested.
The wcf service is hosted in a console application.
In the unittest, i want the possibility to debug this wcf service.
For this, the process has to be attached to the visual studio(2008 in my case) debugger.

You just have to add a reference to EnvDTE, and to add the following pice of code. With that, visual studio attaches the given process to the current debug session.

```
EnvDTE.DTE dte2 =  
(EnvDTE.DTE)System.Runtime.InteropServices.Marshal.GetObject("VisualStudio.DTE.9.0");  
EnvDTE.Debugger debugger = dte2.Debugger;  
foreach (Process program in debugger.LocalProcesses)  
{  
    if (program.Name.Contains("MyApp.exe"))  
    {  
        program.Attach();  
    }  
}
```

It's handy, its easy and there is not a lot of documentation about it...
There are a lot of other EnvDTE(80/90) assemblys, and i guess they are for specific visual studio versions. But i don't see the benefit in using them.

Posted by suntsu in .net at 09:18