Upgrading from .NET 1.1 to .NET 2.0, what to expect?

Asked 16 years, 3 months ago Modified 13 years, 11 months ago

Viewed 4k times

8

I'm working on a big .NET 1.1 project, and there exists a wish to upgrade this, majorily to be able to use better tools like Visual Studio 2008, but also because of the new features and smaller amount of bugs in the .NET 2.0 framework.

The project consist for the bigger part of VB.NET, but there are also parts in C#. It is a Windows Forms application, using various third party controls. Using .NET remoting the rich client talks to a server process which interfaces with a MSSQL 2000 database.

What kind of issues can we expect in case we decide to perform the upgrade?

c# .net vb.net winforms deprecated

Share

Improve this question

Follow

edited Sep 11, 2008 at 19:56

swilliams

48.8k ● 27 ● 102 ● 130

asked Sep 11, 2008 at 19:48

Tobi
81.4k ● 6 ● 34 ● 37

13 Answers

Sorted by: Highest score (default)

7

There is a change to the theading model in .Net 2.0 onwards where unhandled exceptions in a thread will cause the whole app to terminate. I ran into this when updating an app that did lots of threading and occasionally crashed. Obviously the .Net 2.0 model is more robust as you should certainly be catching these anyway, but it was the only really issue I came across when making the migration.

This article talks all about it: http://odetocode.com/blogs/scott/archive/2005/12/14/2618.aspx

Share Improve this answer

Follow

answered Sep 15, 2008 at 10:43

MikeeMike
1,484 ● 1 ● 12 ● 7

5

We're looking at doing the same migration right now Tobi. First, you can get a good idea of what to expect by making a copy of your project (or a portion of it) and give it a "dry run" through the .NET 2.0 compiler. My experience with this was that the 2.0 compiler gives more warnings about bad programming practices that the 1.1 compiler let slide. The compiler will warn you about implicit casts, "ambiguous" return paths (a code path where a function doesn't return a value), and some other minor things.

Here's a few links that you might find helpful: .NET Framework Compatability

Word Document of Breaking changes in .NET Framework 2.0

Share Improve this answer

Follow

answered Sep 11, 2008 at 20:05

Mark

429 ● 1 ● 3 ● 9

2

Nothing, really. You'll find a couple warnings on compilation about obsolete methods, but often those are trivial to fix.

You should shoot big and go for 3.5. The water is niiiiiiice in here.

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:50

user1228

2

Take a peek at this whitepaper on evolving a .NET 2.0 application to 3.5. I hold that the changes from 1.1 to 2.0 are more significant, but the process should be similar.

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:57

Anthony Mastrean

22.4k ● 22 ● 115 ● 190

2

In addition to the app configuration stuff mentioned above, if you use any XSD validation you will need to replace some code around loading and validating XML.

Share Improve this answer

Follow

answered Sep 11, 2008 at 20:09

Kevin Kershaw

196 ● 2 ● 7

1

The most compilation warnings you'll see are if you use app.config to store program settings. The 1.1 configuration class was deprecated for System.Configuration.ConfigurationManager.

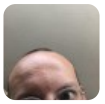
Other warnings you may see coming from the compiler will be for uninitialized variables (set them to "= nothing" or "= null;" in the variable declaration to make them go

away), and unused variables (the compiler is sure they're safe to delete).

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:55



Chris Wenham

24k ● 13 ● 61 ● 70

1

Most of the code should still compile except for a few warnings about stuff being obsolete.

But there are a couple of things you should look out for with respect to Visual Studio generated code.

If you've generated strongly typed datasets in Visual Studio 2003 you can forget about editing them in newer versions of visual studio. You'll have to rebuild them or better just replace them with something like nHibernate for ultimate OR-mapper-bliss

The designer for forms should still work with old forms. You can get some confusion though because 2005 and 2008 use partial classes here. So if you create new forms the code looks different from the old ones. I have never upgraded an ASP.Net application so I don't know about web-forms but I guess it will work the same as winforms stuff. Mostly it will work but expect some designer weirdness.

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:59

Mendelt

37.5k ● 6 ● 75 ● 97

1

.NET 1.1 and .NET 2.0-3.5 are entirely different frameworks, and more importantly, .NET 3.5 is just a set of extra assemblies you can add to your .NET 2.0 project - none of the core assemblies actually got changed, as far as I'm aware - and an upgraded compiler that knows about the syntax sugar called LINQ, extension methods, etc.

In other words, I do not think a .NET 2.0-3.5 upgrade is very similar to a .NET 1.1-2.0 upgrade.

Share Improve this answer

Follow

answered Sep 11, 2008 at 20:12

Tobi

81.4k ● 6 ● 34 ● 37

1

Things will probably compile OK, but we had a few nasty runtime issues with an application we upgraded at the start of the year.

First, we had a number of problems with timezone handling in DateTime objects when calling 1.1 webservices from a 2.0 application, as the conversions to and from UTC when serializing to the wire appeared to work differently between framework versions.

Also, 2.0 async webservices use the klutzy event-based mechanism instead of the IAsyncResult pattern, which is a royal pain if you are batching your requests.

Finally, we had some legacy code that hosted an embedded browser using Microsoft.mshtml.dll. Upgrading to 2.0 caused the application to silently switch to a newer version of that dll, which had some changed behaviour related to javascript interaction. This last one is a bit of an obscure case, but shows that moving to the newer runtime may have implications for any COM interaction you might have.

Hope this helps!

Share Improve this answer

Follow

answered Sep 15, 2008 at 11:26

Russ

1,544 ● 1 ● 9 ● 5

1

The way we were doing email had to change. The 1.1 version used system.WEB.mail, with

```
Imports System.Web.Mail
'
Dim message As New MailMessage' this is a web.mail
Dim objConn As SmtpMail
Dim objAttach As MailAttachment
    '
message .From = "From@us.com"
   ' more properties assigned to objMail
objAttach = New MailAttachment(ExportName)
message.Attachments.Add(objAttach)
   ' Here's where we actually send the thing
SmtpMail.SmtpServer.Insert(0, "127.0.0.1")
objConn.Send(objMail)
```

and the new one has system.NET.mail

```
         Imports System.Net.Mail
        '
         Dim message as MailMessage ' this is a net.ma
msg
         Dim data As Attachment
        Dim client As New SmtpClient("127.0.0.1")
    '
        data = New Attachment(ExportName)
    ' Create the message and add the attachment
        message = New MailMessage(EmailFrom, EmailTo,
        message.Attachments.Add(data)
' Send the message
        client.Send(message)
```

Share Improve this answer

Follow

answered Sep 16, 2008 at 16:58

CindyH

3,026 ● 2 ● 25 ● 38

RESX files upgrade problems

1

Watch out for internationalized RESX files.

When you reopen a ,net 1.1 form in .net 2.0 the RESX file gets upgraded to a new version. In .net 1.1 the foreign language .resx file only contained the changes. In .net 2.0 ALL of the fields in the default .resx file now get moved into the foreign language resx file. (.fr.resx for example). If you have already internationalized the form all of the foreign language resx files will have to be looked at.

Internationalisation Tools

Some tools that you may have used/written yourself to do internationalization en mass may not work anymore as they may have used numbered resources. (Multi Lang & Infragistics)

Infragistics Winforms controls modify the InitializeForm() in .net 1.1 and access resources using a resource numbering system. When migrated to .net 2.0 the numbering of the Infragistics resources will fail as the resx file is regenerated. You will need to upgrade your Infragistics libraries.

Share Improve this answer

Follow

edited Oct 5, 2008 at 8:28

community wiki
2 revs
user18443

0

You probably won't have any **breaking** issues, though you may get some deprecated method warnings. The compiler should generally tell you what the replacement is though. I know that some of the System.Configuration things were updated.

Share Improve this answer

Follow

answered Sep 11, 2008 at 19:55

swilliams
48.8k ● 27 ● 102 ● 130

0

There shouldn't be too much of a problem as in theory it is backward compatible (I note MikeeMike's comment about thread exceptions). After the move, you'll fine there are quite a few nice things like generics. Although you don't want to port all your collections to generics in one fell swoop, once you've done this your code should be more reliable due to the reduced number of casts - and probably quicker (although mileage may vary in this aspect). At the moment I'm about to start the .NET 2 -> .NET 4 conversion of three of my products. The main advantage will be the further improvements in multi-threaded support (parallel foreach loops, etc).

Share Improve this answer

Follow

answered Jan 4, 2011 at 14:50

winwaed

7,791 ● 6 ● 42 ● 84