

Re: Explicit Linking of DLL's in VB.net

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2007-02/msg00167.html>

- *From:* "Michel Posseth [MCP]" <MSDN@xxxxxxxxxxxx>
 - *Date:* Fri, 2 Feb 2007 07:16:09 +0100
-

Btw, for the previous post, it is a vb.net dll. Aka managed, I believe....? (can you tell I am a bit new to this?)

yes this technique will only work on managed assemblies (exe or dll)

I do have one question though. The "YourObject" is a form and is limited to the form methods. How can I send information and and forth from the main program to the external dll?

Wel implement propertys , methods , events in your interface and start communicating :-)

i thought this is the easiest way , however it is possible to send parameters to the constructor (in the CreateInstance method) however i didn't bother as i communicate through my interface with the object (send parameters through properties, or methods , receive messages back through events)

regards

Michel

"Noone" <None@xxxxxxxx> schreef in bericht
<news:p4g4s25pn4e4ne221rocm7kc1norva6ln7@xxxxxxxx>

Hah... Luckily it is looking like I am doing something right. With further research that is exactly the way I went. I found I could load a form exactly how you did it. I do have one question though. The "YourObject" is a form and is limited to the form methods. How can I send information and and forth from the main program to the external dll? Any ideas? Right now I am limited to form.show, form.height, etc etc. Nothing that I create...

Re: Explicit Linking of DLL's in VB.net

Thanks!

Btw, for the previous post, it is a vb.net dll. Aka managed, I believe....? (can you tell I am a bit new to this?)

On Thu, 1 Feb 2007 06:37:01 -0800, Michel Posseth [MCP]
<MichelPossethMCP@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Hello Noone

The concept is pretty simple

1. create a generic interface and compile this to a dll
2. create your plugin and implement the interface in the class you want to start from the outside
3. create a application and set a reference to the interface dll

now you do

```
Dim objAssembly As Reflection.Assembly  
objAssembly =  
Reflection.Assembly.LoadFrom(FullPathToAssemblyDllOrExe)
```

```
Dim YourObject as Yourinterface =  
DirectCast(objAssembly.CreateInstance(Namespace.YourClassToInvoke),  
Yourinterface)
```

Note :
Namespace.YourClassToInvoke (namespace defaults to the assembly name
but
can be set under project , properties , application , root namespace)

And that`s it !! :-)

YourObject is now initiated and can be controled from your code with the
interface that you provided

if You need anny more help feel free to ask (i can create a small demo
for
you and upload it to my server for you to download)

Regards

Michel Posseth

Re: Explicit Linking of DLL's in VB.net

"Noone" wrote:

Thank you very much for the post but I have no idea what you are doing. I have never worked with any of the "Reflection" classes. I am a bit newer to VB.net so I think I am missing something. Can you explain what is going on in more detail and less Dutch? hehe... :)

Thanks,

Josh

On Wed, 31 Jan 2007 21:15:08 +0100, "Michel Posseth [MCP]" <MSDN@xxxxxxxxxxxx> wrote:

Well i did this with an interface

here is my reallife code from a wotking project (sorry comments are for my co workers and they are Dutch)

```
'-----  
'_____  
'---- clsGetObjectFromFile  
'_____  
'---- Purpose : start returns an initiated  
IiQueuObject  
'_____  
'---- Made by : Michel Posseth [MCP]  
'---- Date : 14-11-2006  
'---- Revisions : 21-11-2006 : AssPath  
ingebouwd voor flexibiliteit
```

Re: Explicit Linking of DLL's in VB.net

'____

'-----

Option Strict On

Option Explicit On

Imports System.IO

Public Class clsGetObjectFromFile

''' <summary>

''' Laad een object by zijn assembly naam en
class naam

''' het te laden object moet een IiQueuObject
interface bezitten

''' </summary>

''' <param
name="vstrAssemblyName">Name of the
VSTR assembly.</param>

''' <param name="vstrClassName">Name of
the VSTR class.</param>

''' <returns></returns>

Public Shared Function

LoadMeByName(ByVal vstrAssemblyName
As String, _

ByVal vstrClassName As String) As
IiQueuObject

'<21-11-2006 MP>

If
vstrAssemblyName.StartsWith("[AssPath]")
Then

Dim appath As String =
System.Reflection.Assembly.GetExecutingAssembly.Location

vstrAssemblyName =
vstrAssemblyName.Replace("[AssPath]",
"")

Re: Explicit Linking of DLL's in VB.net

```
vstrAssemblyName = Path.Combine(appath,
vstrAssemblyName)

End If

'</21-11-2006 MP>

Dim objAssembly As Reflection.Assembly

If Not
My.Computer.FileSystem.FileExists(vstrAssemblyName)
Then

ServMain.WriteLogentry("Assembly niet
aanwezig : " & vstrAssemblyName,
EventLogEntryType.Error)

'de assembly is niet aanwezig op deze
lokatie

Return Nothing

End If

objAssembly =
Reflection.Assembly.LoadFrom(vstrAssemblyName)

'voer een cast uit naar ista.IIQueueObject
interface

Try

LoadMeByName =
DirectCast(objAssembly.CreateInstance(vstrClassName),
ista.IIQueueObject)

Catch ex As Exception

ServMain.WriteLogentry("Assembly bezit
niet de juiste interface : " &
vstrAssemblyName,
EventLogEntryType.Error)

Return Nothing

End Try

If LoadMeByName Is Nothing Then

Dim msg As String
```

Re: Explicit Linking of DLL's in VB.net

```
msg = "Assembly : " & vstrAssemblyName  
& Environment.NewLine & _
```

```
"Type : " & vstrClassName & " is niet  
gestart om onduidelijke reden"
```

```
ServMain.WriteLogentry(msg,  
EventLogEntryType.Error)
```

```
'geen error gewoon niets terug geven
```

```
'wanneer we dus iets anders hebben als een  
Nothing pointer
```

```
'dan hebben we een geinitialiseerd object
```

```
Return Nothing
```

```
End If
```

```
End Function
```

```
End Class
```

here is my generic interface

```
'-----
```

```
'----
```

```
'---- Purpose : Provide an generic interface  
for the Queu
```

```
'----
```

```
'---- Made by : Michel Posseth [MCP]
```

```
'---- Date : 20-11-2006
```

```
'---- Revisions :
```

```
'----
```

```
'---- Remarks : Do not break the interface  
signature !!
```

Re: Explicit Linking of DLL's in VB.net

'--- you may extend but never remove
property`s or methods

'---

'-----

Option Strict On

Option Explicit On

Public Interface IiQueueObject

Sub StartProcessing()

Property Parameters() As String

Property ProcessId() As String

Event eFinished(ByVal ProcessId As String,
ByVal msg As String)

Event eError(ByVal ProcessId As String,
ByVal msg As String)

Event eProcessCancelled(ByVal ProcessId
As String, ByVal msg As String)

Property CancellProcess() As Boolean

End Interface

now i can just use anny object like this

'het object welke we gaan starten

'OTask wordt gedefinieerd as
ista.IiQueueObject

'we kunnen dus ieder mogelijk object
opstarten indien het deze
interface
heeft geïmplementeerd

Dim oTask As ista.IiQueueObject =
clsGetObjectFromFile.LoadMeByName(Dr.Item("AssemblyNaam").ToString,
Dr.Item("AssemblyType").ToString)

If Not IsNothing(oTask) Then

Re: Explicit Linking of DLL's in VB.net

```
otask.StartProcessing()
```

```
end if
```

HTH

Michel Posseth [MCP]

"Noone" <None@xxxxxxxx> schreef in
bericht

news:uos1s2h10bh55is17f4oij97cti5nghk2u@xxxxxxxxxxx

Sorry, not invoke, I meant
Interop. I think... Heeelp! :)

On Wed, 31 Jan 2007
19:45:41 GMT, Noone
<None@xxxxxxxx> wrote:

Hello all,

Ok, I want
to create a
program
that will
load plugins
(dll's) from
a
plugin
folder. I can
create the
forms and
put them
into a dll
but I
cannot

Re: Explicit Linking of DLL's in VB.net

actually add
them
dynamically
at run time.
I have tried
to
use
the
LoadLibrary
and GetProc
functions,
which sort
of worked. I
got
the pointer
to the
function but
I cannot
actually
RUN the
function
like I can in
C++. I have
heard some
things about
Invoking(?)
I
believe but I
cannot find
enough
about it.
Any
comments
or
suggestions
would be
EXTREMELY
helpful.

Thanks,

Josh

