

Re: cyclic dependency

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-02/msg01890.html>

- *From:* "Dave" <Dave@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 9 Feb 2006 06:28:31 -0800
-

I also have a problem with circular dependency which I would like some advice about. I have to say I don't understand the reasoning behind this circular dependency rule at all. In the old days you could have two classes that referred to each other and the linker never complained, what's the difference? Anyway, I have a perfectly reasonable situation, I have a configuration class that contains information about my system generally, and I have an event logging class that writes events to a file. They are in separate assemblies, which seems reasonable. The configuration class contains a log file retention period which the event logger needs. However the configuration class also needs to log events. Can't be done. Now, I could put both classes in the same assembly. In fact I could put all my classes into one great big assembly, but surely that is not the way. What is the answer?

--
Dave

"Nicholas Paldino [.NET/C# MVP]" wrote:

Po,

Forgive me for saying so, but I see cyclical references between assemblies as a sign of bad design. If they really have a need to have that level of familiarity with each other, then I don't see why they can't be in the same assembly.

If you really need this, the only way to do it is the way that Jay suggested, using interfaces. Otherwise, you will have to place the classes in the same assembly.

On top of that, I don't think that you will find circular references in ..NET anytime soon. I think it was considered at one point, but shot down pretty quickly.

Hope this helps.

--

Re: cyclic dependency

```
public int CalBalls(int i)
{ // return some calculation as int }
}
}
```

"Jay B. Harlow [MVP – Outlook]" <Jay_Harlow_MVP@xxxxxxxxxxxxxx> wrote in message news:uX7Dn%238BGHA.2036@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Pohihihi,
Normally when I have two projects that need to refer to each other I use a Separated Interface Pattern.

<http://www.martinfowler.com/eaCatalog/separatedInterface.html>

Depending on the needs of the solution will decide if I use 2 or 3 projects to implement the Separated Interface Pattern.

The "easiest" way is to put the interfaces in the first project, then reference the first project from the second. Alternatively you can put the interfaces in a project by themselves, and reference the interface project from both of the other projects.

Something like:

```
' Project A

Public Interface InterfaceA
Sub SomeMethod()
End Interface

Public Class Something

Sub DoSomething(anA As InterfaceA)
anA.SomeMethod()
End Sub

End Class

' Project B
' references Project A

Public Class SomethingElse
Implements InterfaceA

Sub DoSomething()
Dim something As New Something
something.DoSomething(Me)
```

Re: cyclic dependency

End Sub

Sub SomeMethod() Implements InterfaceA.SomeMethod

...

End Sub

End Class

--

Hope this helps

Jay [MVP – Outlook]

.NET Application Architect, Enthusiast, & Evangelist

T.S. Bradley – <http://www.tsbradley.net>

"Pohihihi" <noemail@xxxxxxxxxxxx> wrote in message
news:edk95d5BGHA.1544@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Hello,

I have 2 project A & B belonging to same solution. Now what I want is that few forms I am accessing from A in B and few I am doing that other way around. What should I do to not get the error while building solution, I mean by the build sequencing.

Thanks,

Po