

How do I create partial class to be nested under the main class file?

How do I create partial class to be nested under the main class file?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-03/msg04439.html>

- *From:* "Joseph Geretz" <jgeretz@xxxxxxxxxxx>
 - *Date:* Wed, 28 Mar 2007 14:41:27 -0400
-

When I create a Form, the VB IDE creates the following files in the following hierarchy:

```
Form1.cs
Form1.Designer.cs
Form1.resx
```

Both Form1.cs and Form1.Designer.cs are partial implementations of a single physical class definition. The VB IDE properly recognizes Form1.cs as a Form (i.e. double-clicking opens the Form designer). It also recognizes Form1.Designer.cs as additional code implementation. It properly nests this class file under Form1.cs and double-clicking the file in the IDE opens to a code window, rather than a Form design window. So far so good.

I'd like to extend the Form interface with a bunch of my own methods. So I create a new class, Form1.Interface.cs, defined as follows:

```
partial class Form1
{
internal void DoThis()
{
}
internal void DoThat()
{
}
```

I'd like Form1.Interface.cs to behave just as Form1.Designer.cs does. That is, I'd like this class to be nested beneath Form1.cs to make it obvious that Form1.Interface.cs is a partial class addition to the partial class defined in Form1.cs. I'd also like this class to open in the IDE just as Form1.Designer.cs does; that is, double-clicking this module should open a code window, not a Form design window.

But I can't get either of these to happen. My module Form1.Interface.cs is treated by the IDE as though it is a Form. It doesn't appear nested under Form1.cs and double-clicking it opens a Form design windows which is totally inappropriate since this module is not intended to implement any visual form

How do I create partial class to be nested under the main class file?

How do I create partial class to be nested under the main class file?

elements at all!

(BTW, the compiler properly aggregates this into a single Form1 class implementation. So there's no problem as far logical software construction is concerned. However, it would be nice if I could get the IDE to recognize the relationship between Form1.cs and Form1.Interface.cs and to properly deal with this relationship (i.e. nesting and default open action) just as it does with Form1.cs and Form1.Designer.cs.)

What am I doing wrong?

Thanks for your help!

– Joseph Geretz –

.