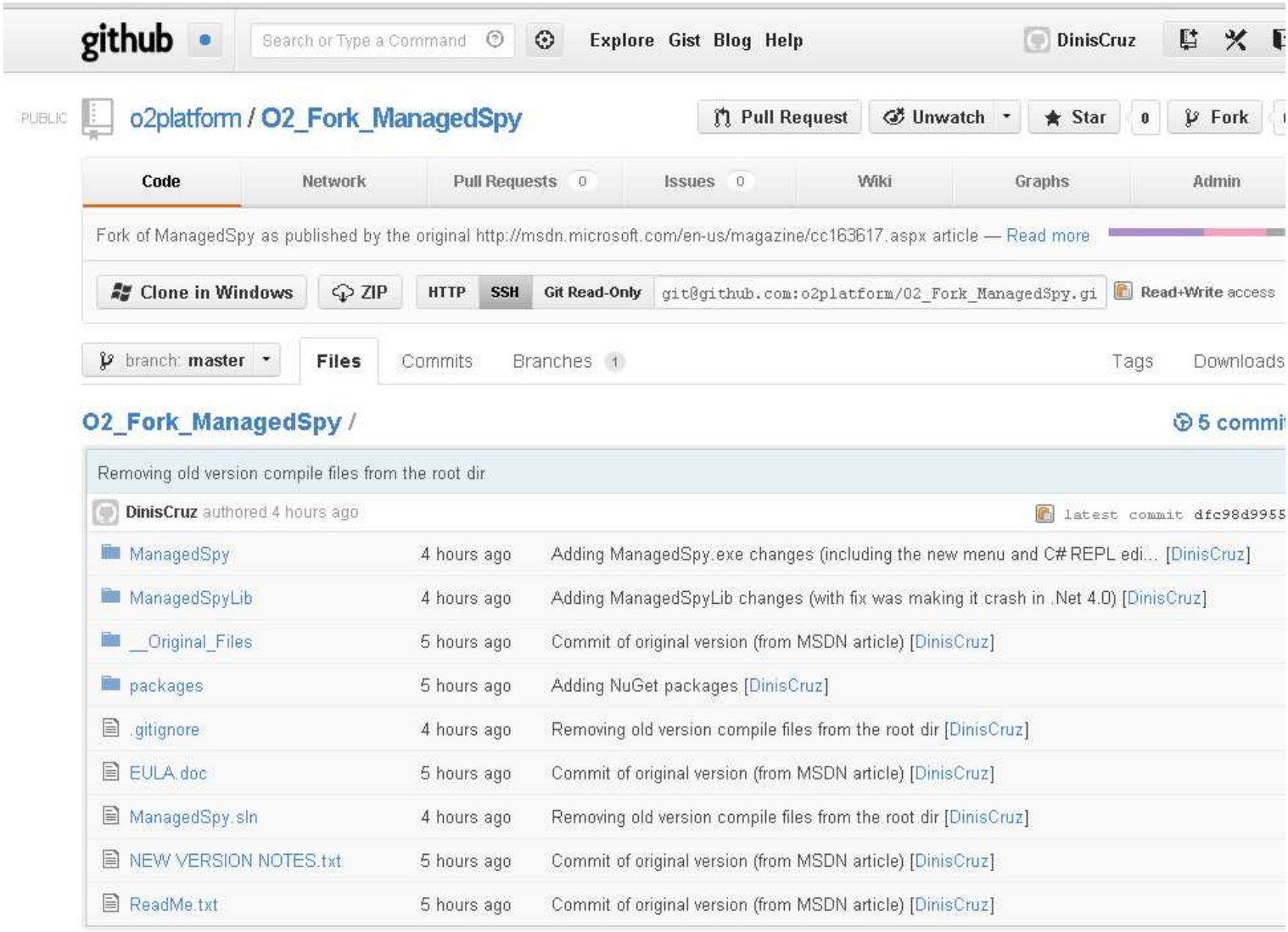
ManagedSpy - Adding new features to ManagedSpy.exe (part 2)

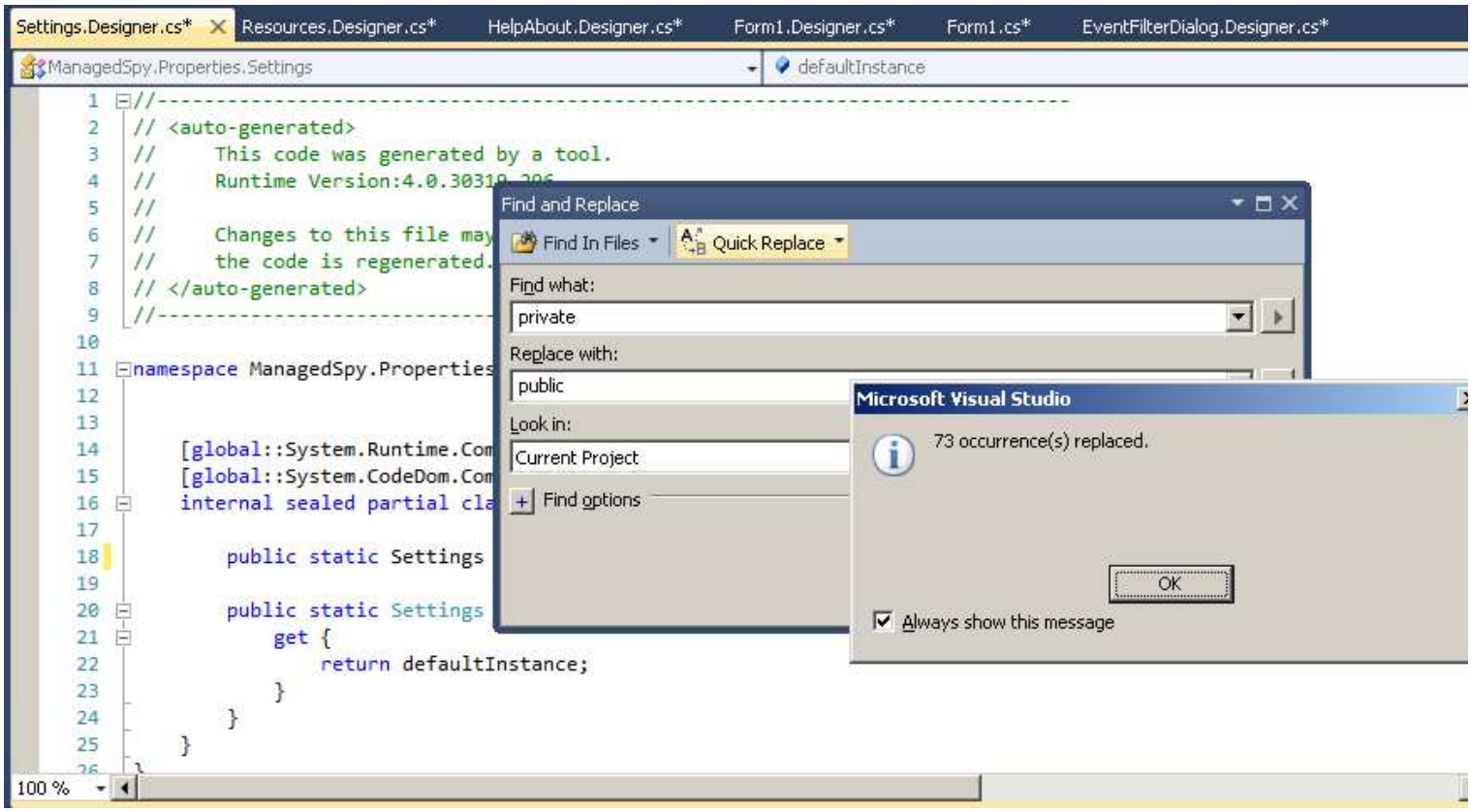
Before we go much further lets move this code to Git

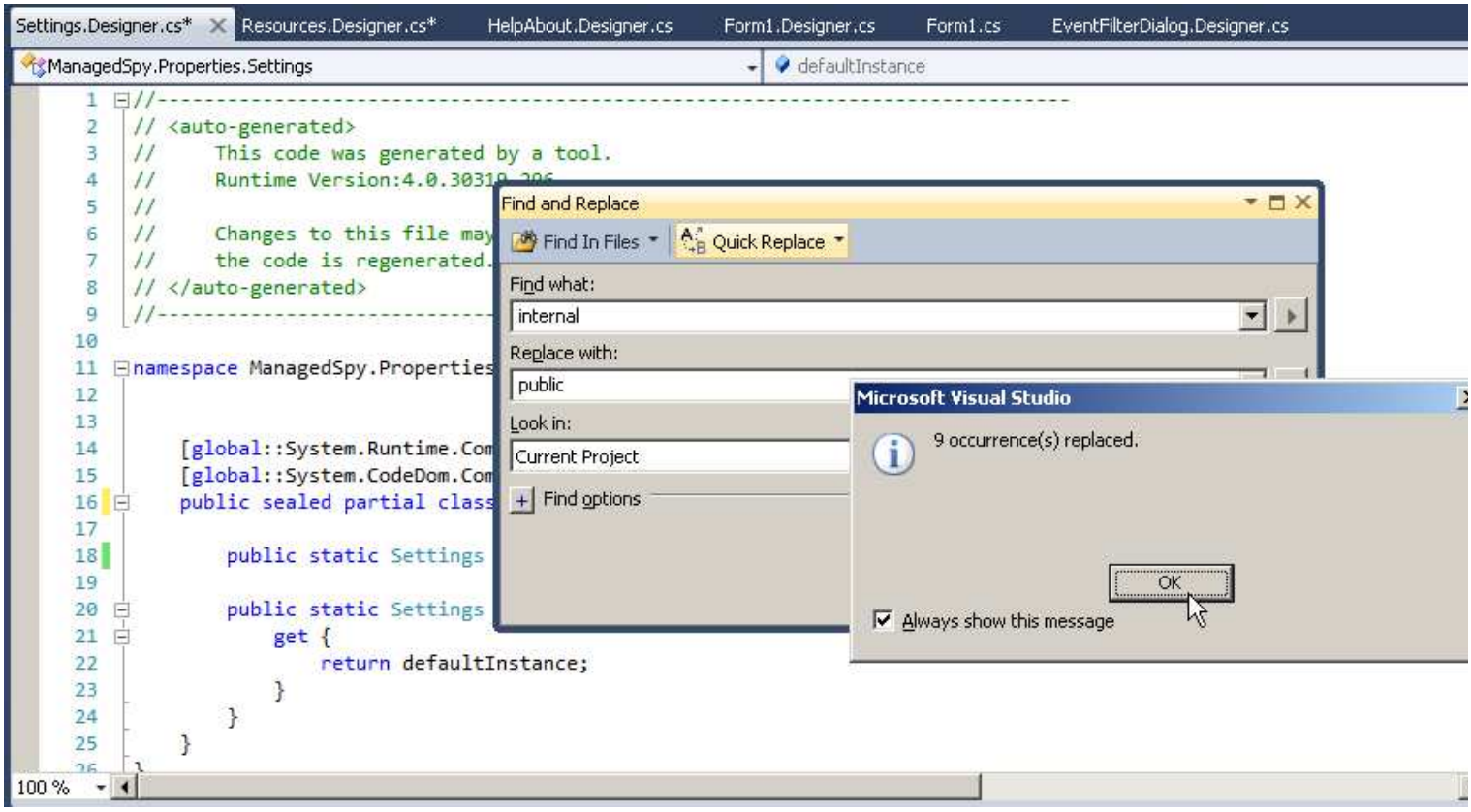
see file: *"ManagedSpy - Moving Original code (and some changes) to GitHub.pdf"* for how it was done

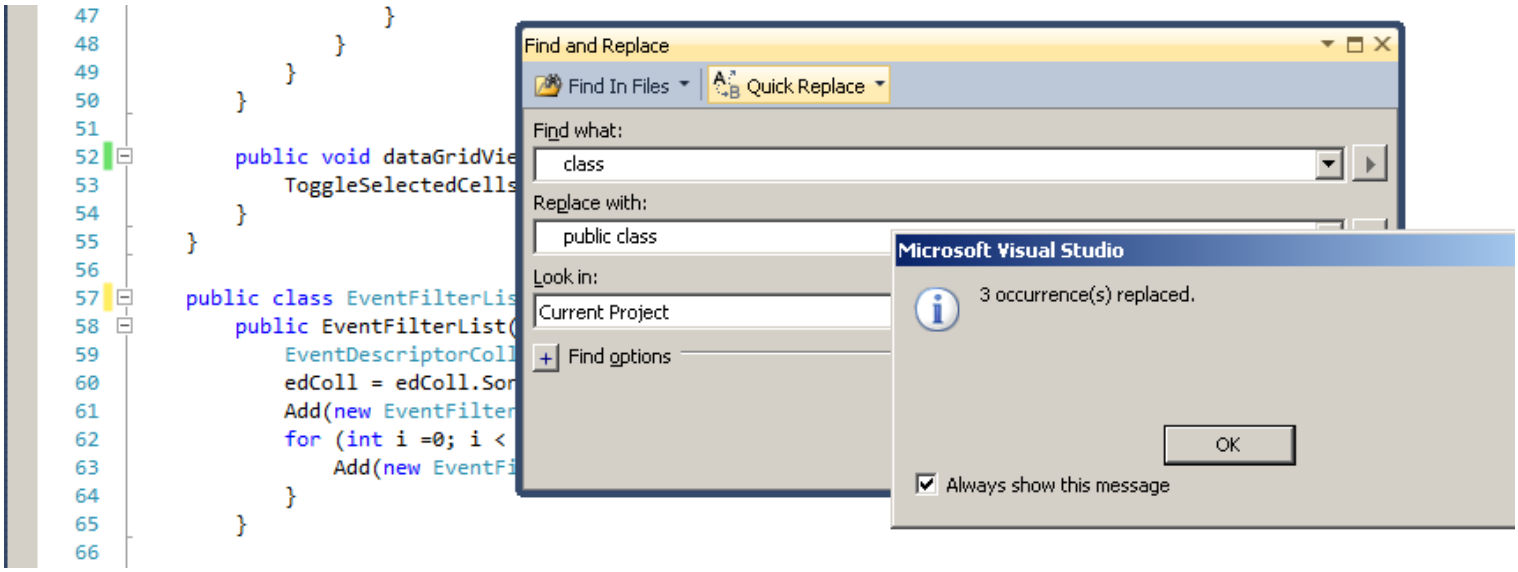


Making all classes and enums public

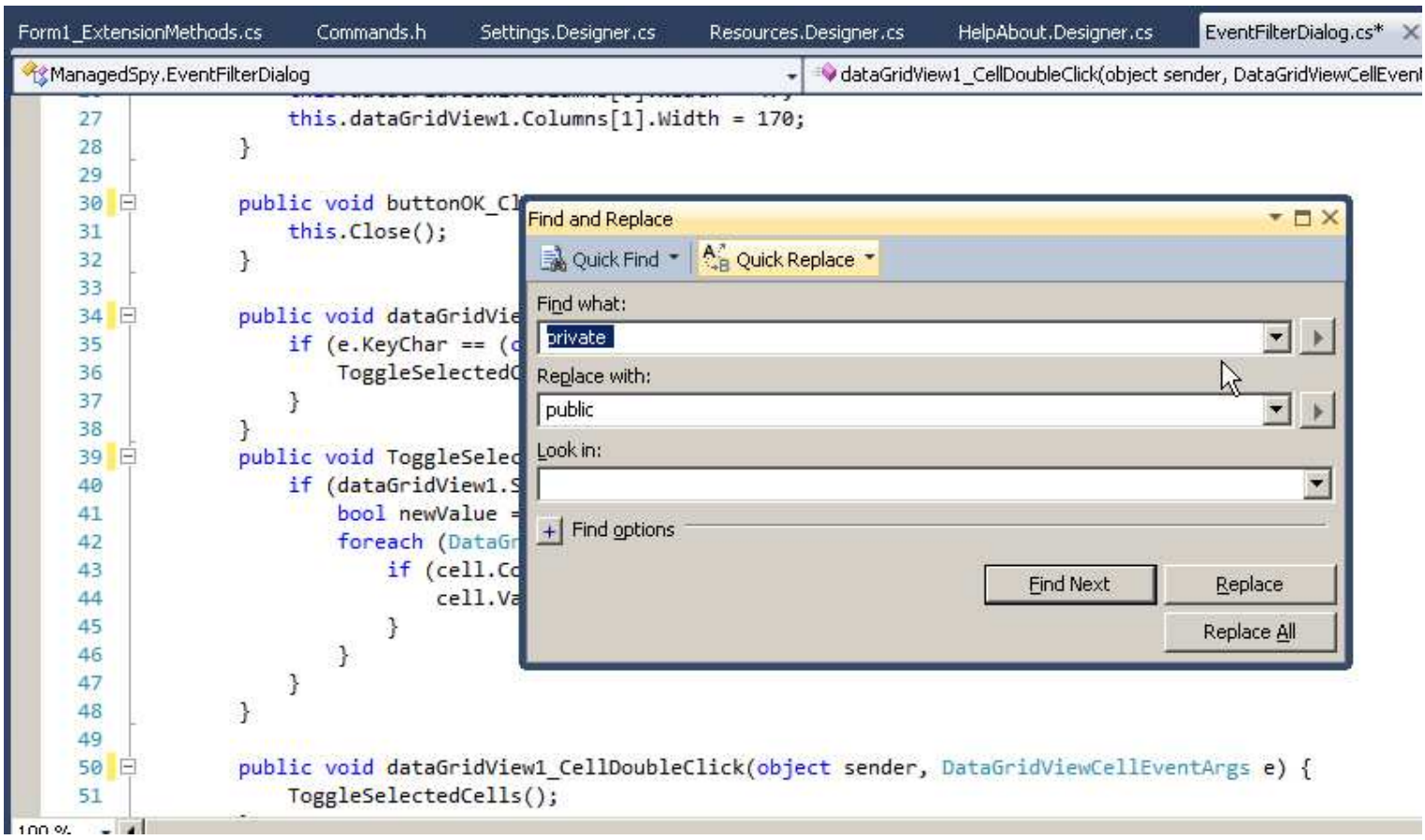
which will make the ManagedSpy.exe objects and classes easier to consume from the REPL environment


Settings.Designer.cs*
Resources.Designer.cs*
HelpAbout.Designer.cs*
Form1.Designer.cs*
Form1.cs*
EventFilterDialog.Designer.cs*
ManagedSpy.Properties.Settings
defaultInstance
Find and Replace
Find In Files
Quick Replace
Find what:
private
Replace with:
public
Look in:
Current Project
Find options
Microsoft Visual Studio
73 occurrence(s) replaced.
OK
Always show this message
100 %

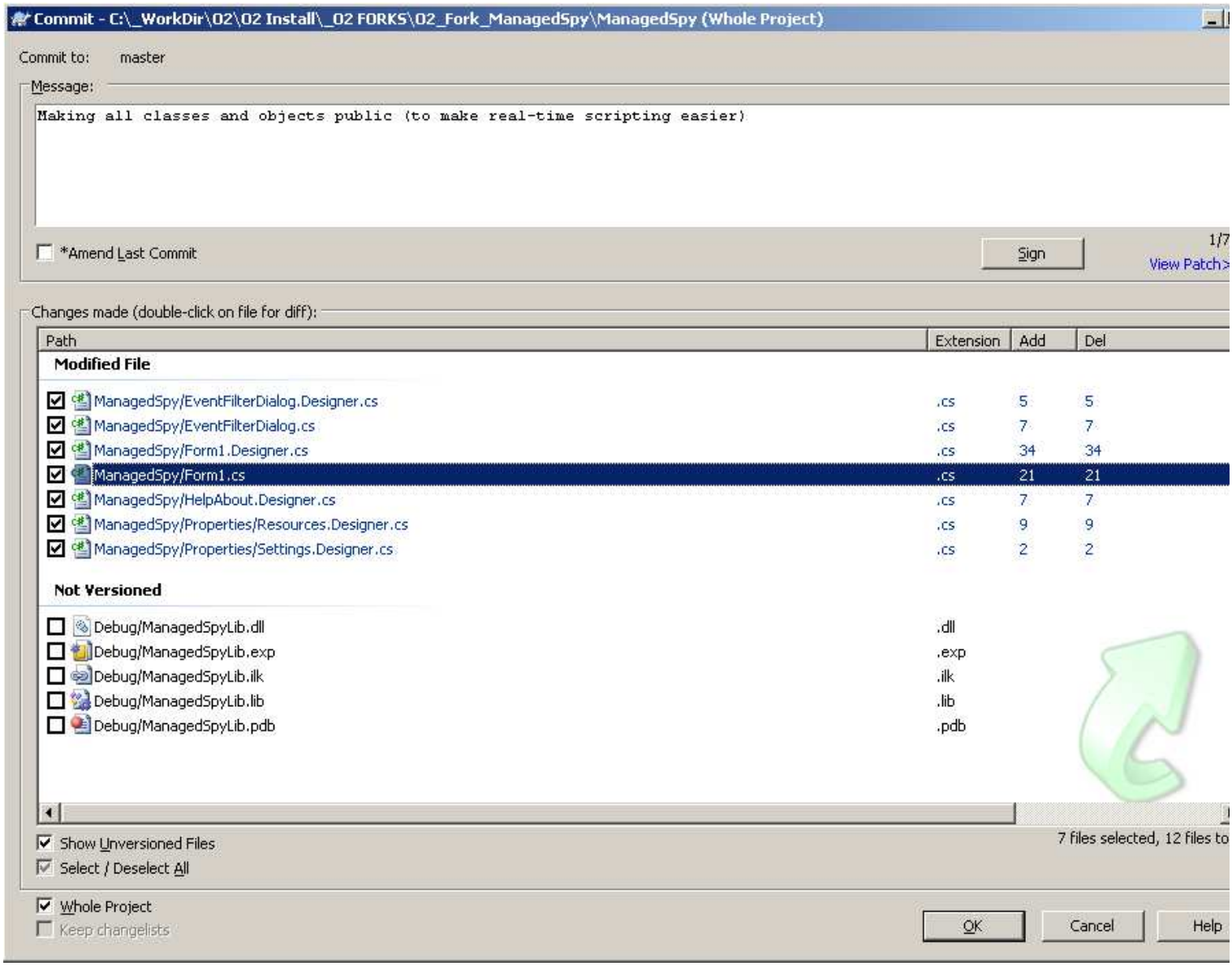

Settings.Designer.cs*
Resources.Designer.cs*
HelpAbout.Designer.cs
Form1.Designer.cs
Form1.cs
EventFilterDialog.Designer.cs
ManagedSpy.Properties.Settings
defaultInstance
Find and Replace
Find In Files
Quick Replace
Find what:
internal
Replace with:
public
Look in:
Current Project
Find options
Microsoft Visual Studio
9 occurrence(s) replaced.
OK
Always show this message
100 %



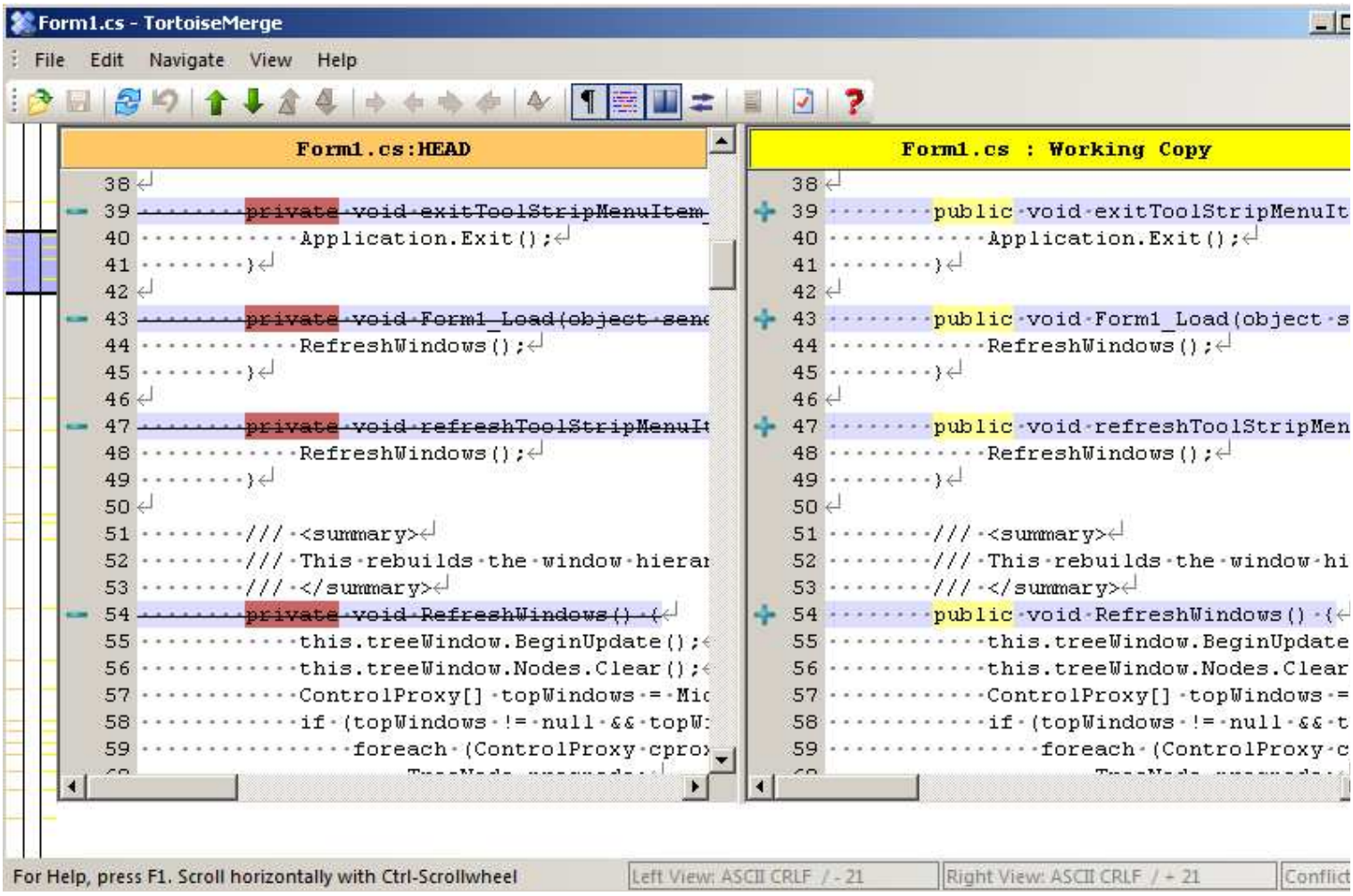
To make it easier to script, lets make app private methods public



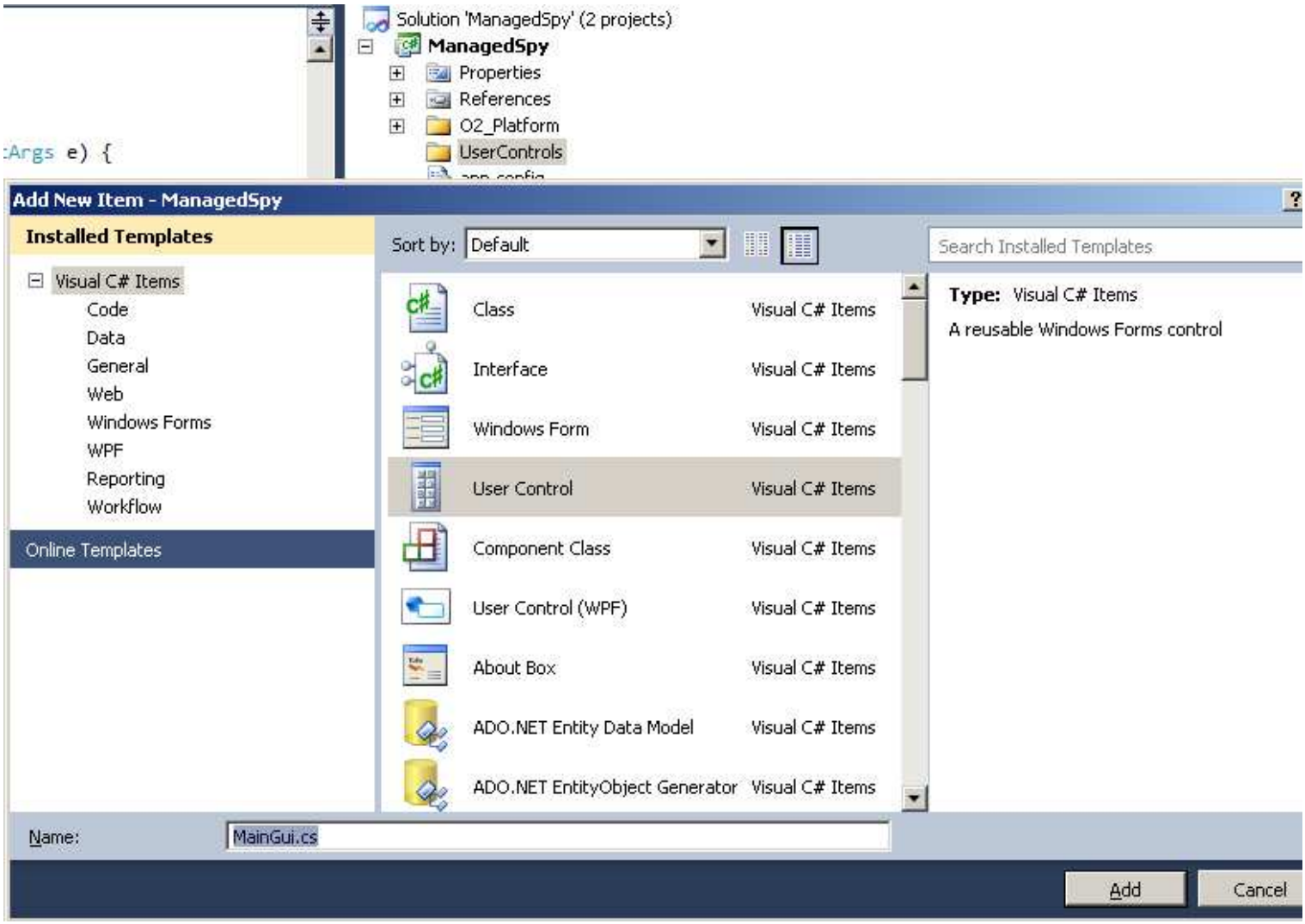
Next commit the changes (which are all the private to public replaces)


Commit - C:_WorkDir\02\02 Install_02 FORKS\02_Fork_ManagedSpy\ManagedSpy (Whole Project)
Commit to: master
Message:
Making all classes and objects public (to make real-time scripting easier)
*Amend Last Commit
Sign
1/7
View Patch >
Changes made (double-click on file for diff):
Path
Extension
Add
Del
Modified File
ManagedSpy/EventFilterDialog.Designer.cs .cs 5 5
ManagedSpy/EventFilterDialog.cs .cs 7 7
ManagedSpy/Form1.Designer.cs .cs 34 34
ManagedSpy/Form1.cs .cs 21 21
ManagedSpy/HelpAbout.Designer.cs .cs 7 7
ManagedSpy/Properties/Resources.Designer.cs .cs 9 9
ManagedSpy/Properties/Settings.Designer.cs .cs 2 2
Not Versioned
Debug/ManagedSpyLib.dll .dll
Debug/ManagedSpyLib.exp .exp
Debug/ManagedSpyLib.ilk .ilk
Debug/ManagedSpyLib.lib .lib
Debug/ManagedSpyLib.pdb .pdb
Show Unversioned Files
Select / Deselect All
7 files selected, 12 files to
Whole Project
Keep changelists
OK
Cancel
Help

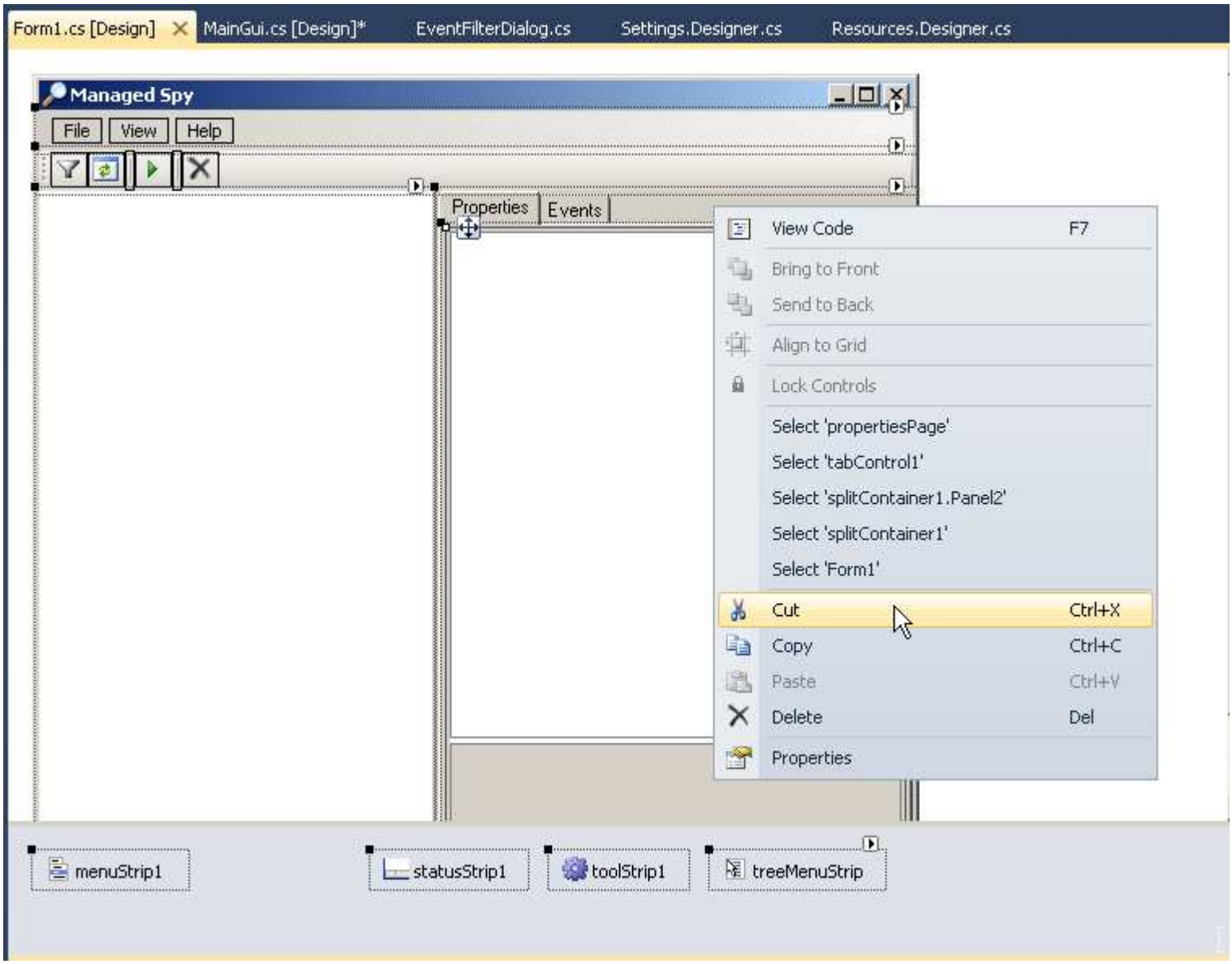
For example:



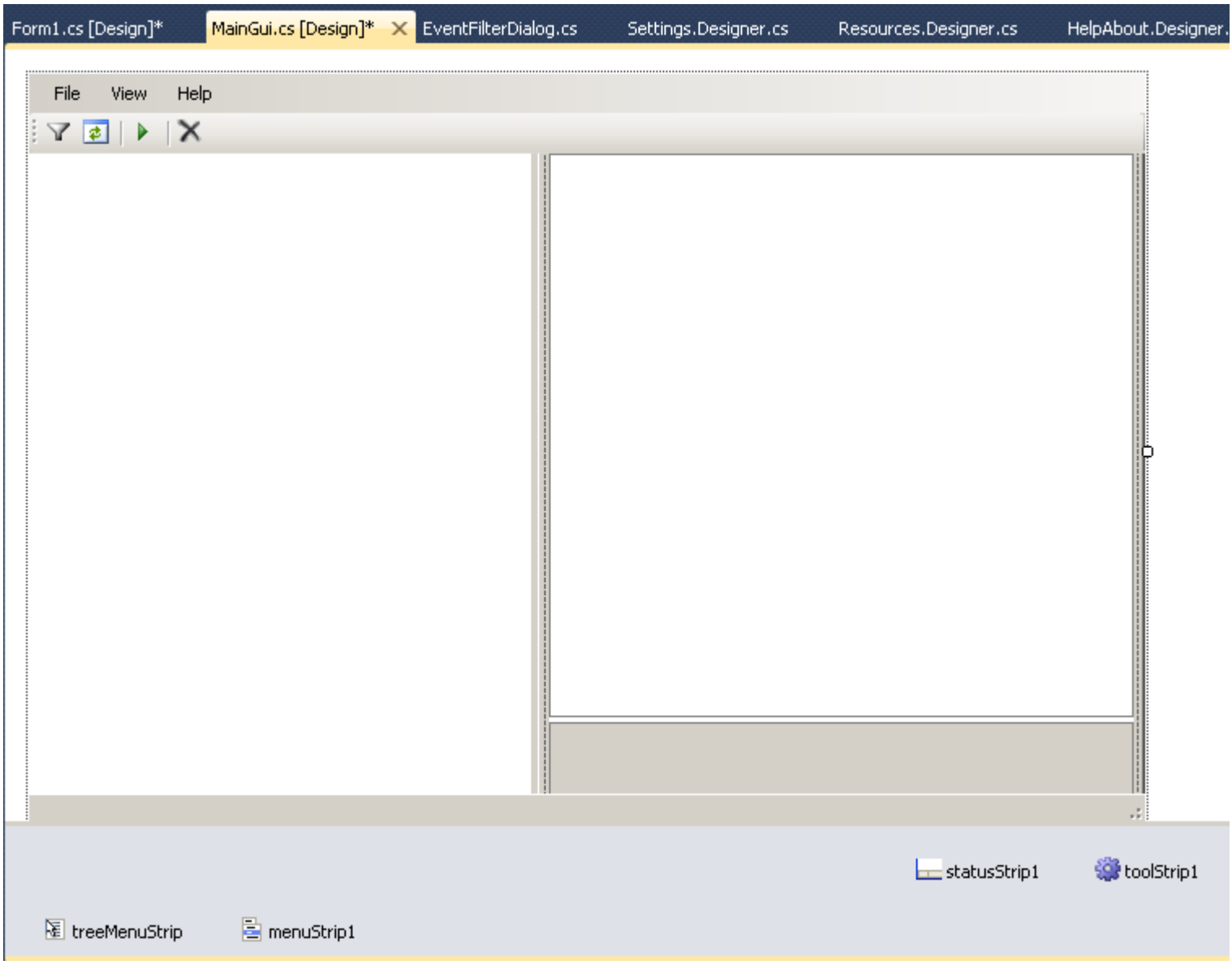
To help with debugging and modularity, let's move the main Form1 code into user control:



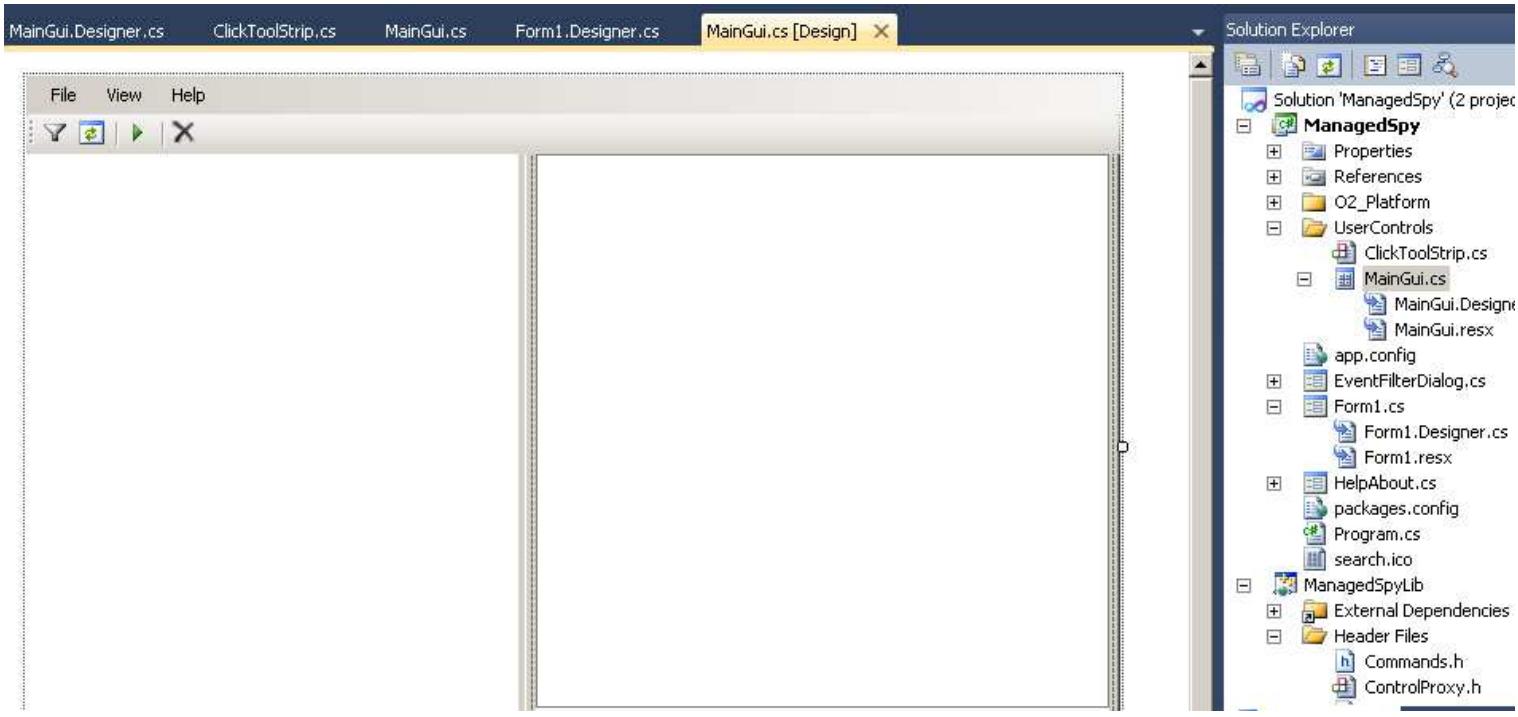
Cut all controls from In *Form1.cs*



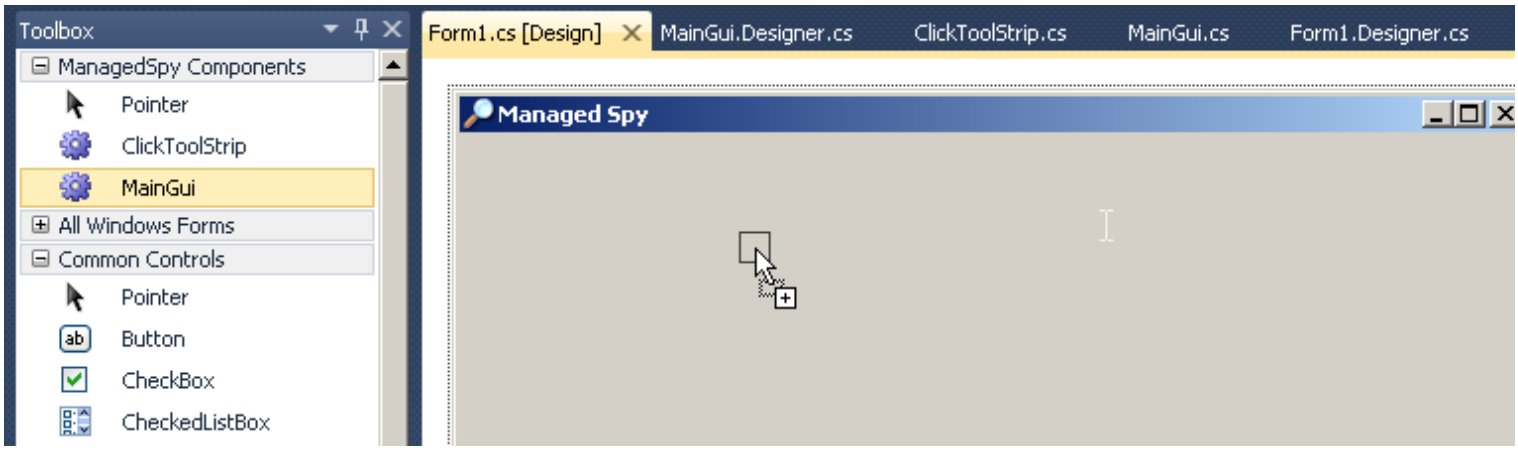
and Paste them into *MainGui.cs*



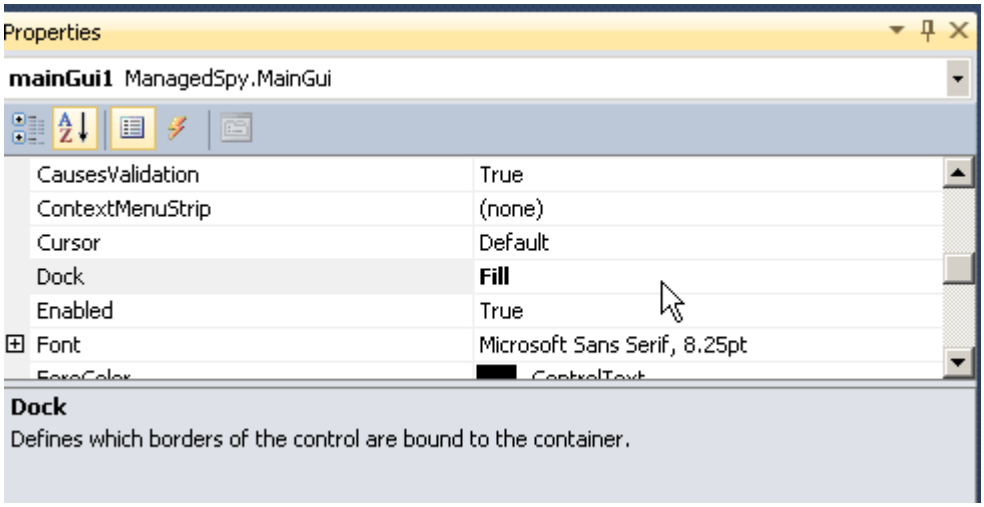
After a little bit of refactoring we now have the original ManagedSpy as UserControl



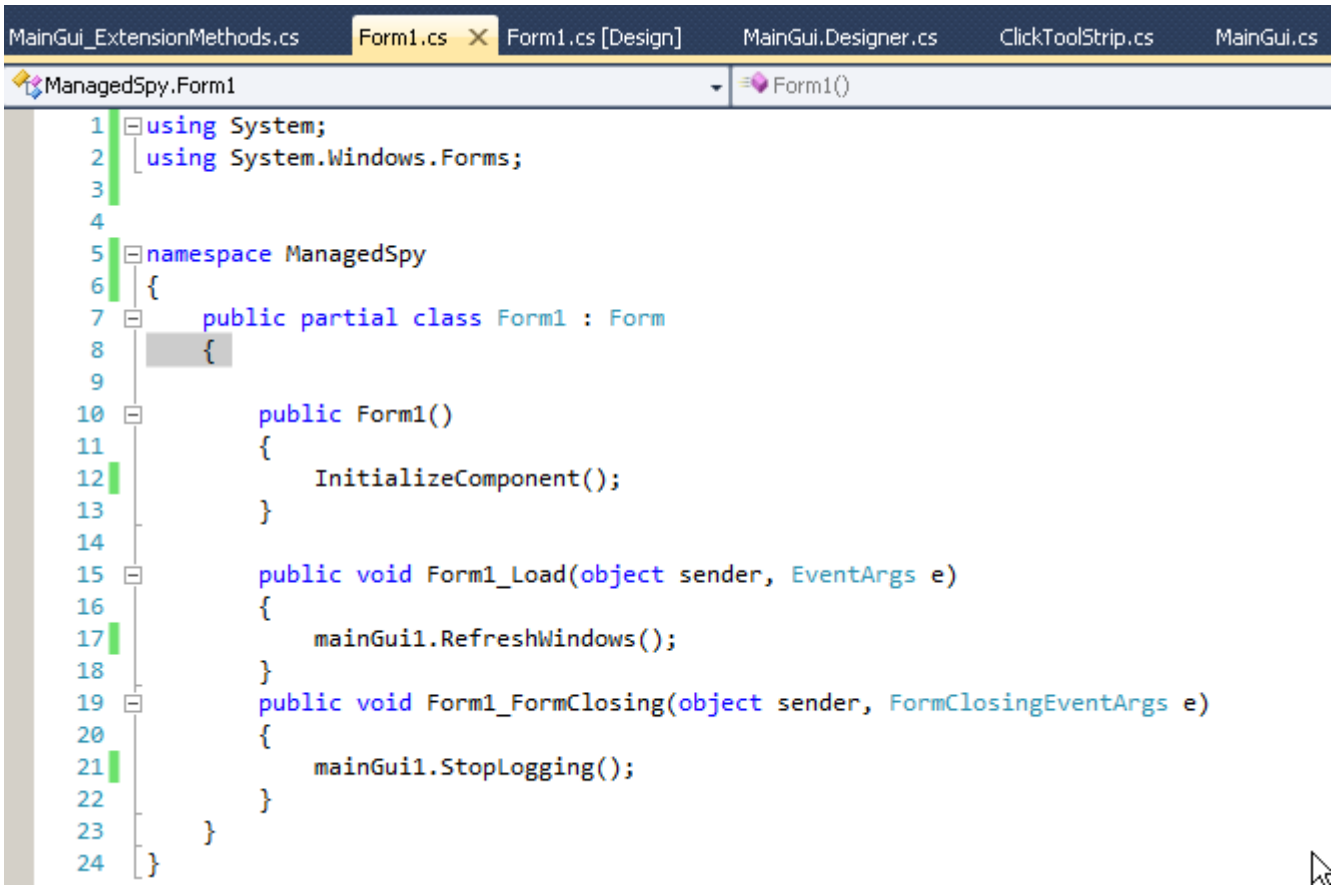
which is available on the ToolBox's *ManagedSpy Components* and can be added to Form1 via Drag-n-drop



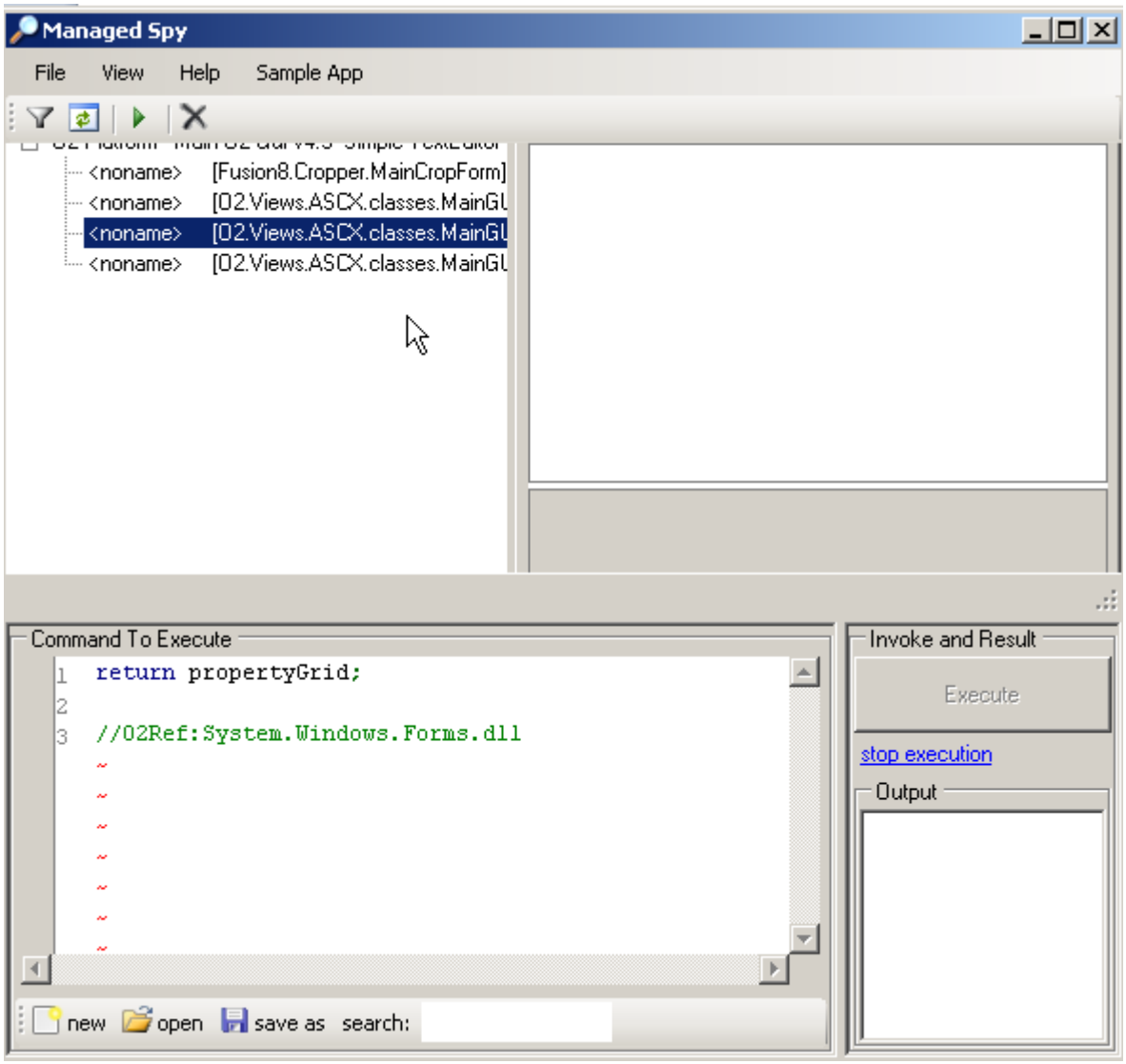
Set the *Dock* property to *Fill*



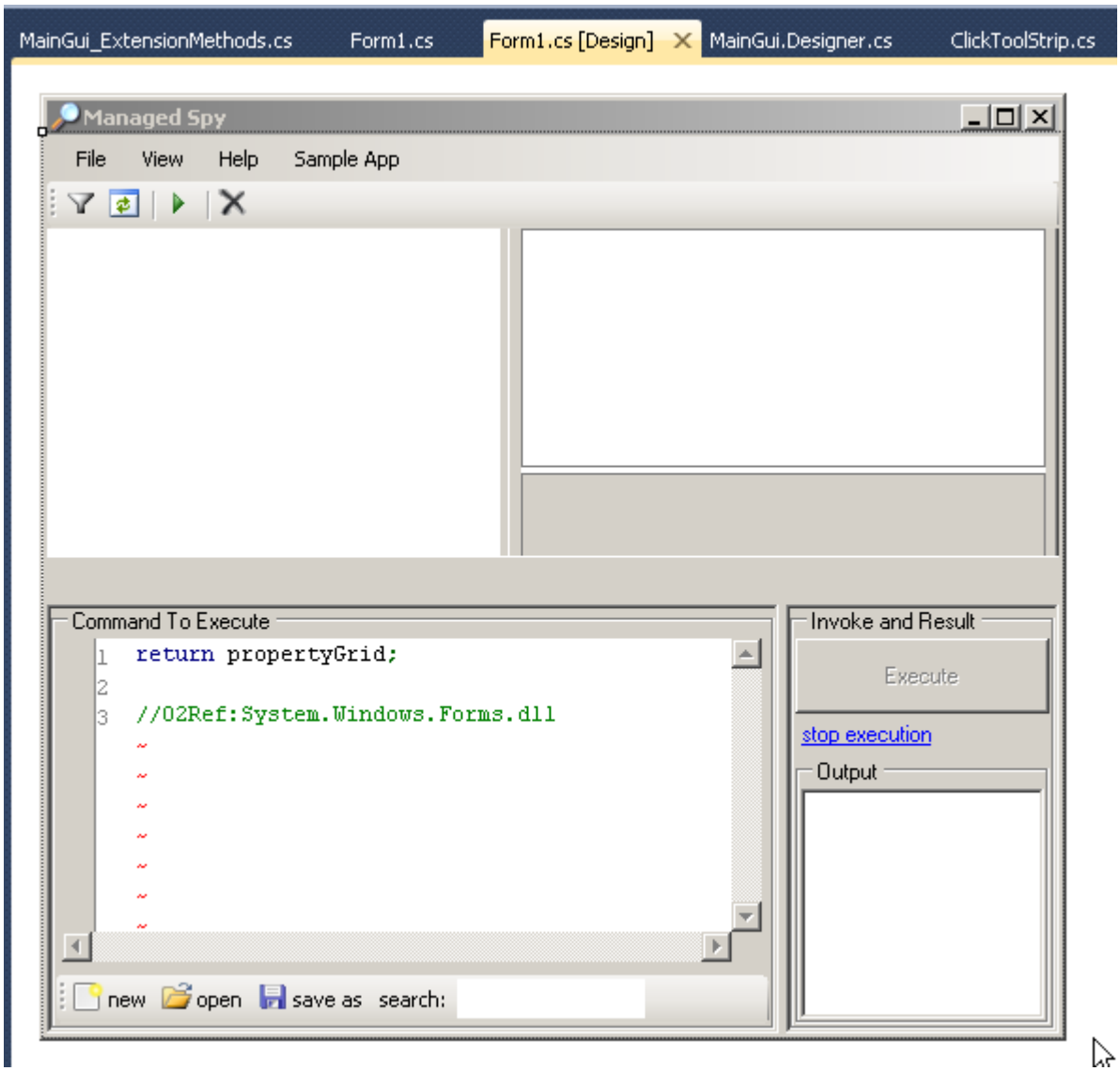
And make a couple final refactorings to Form1.cs



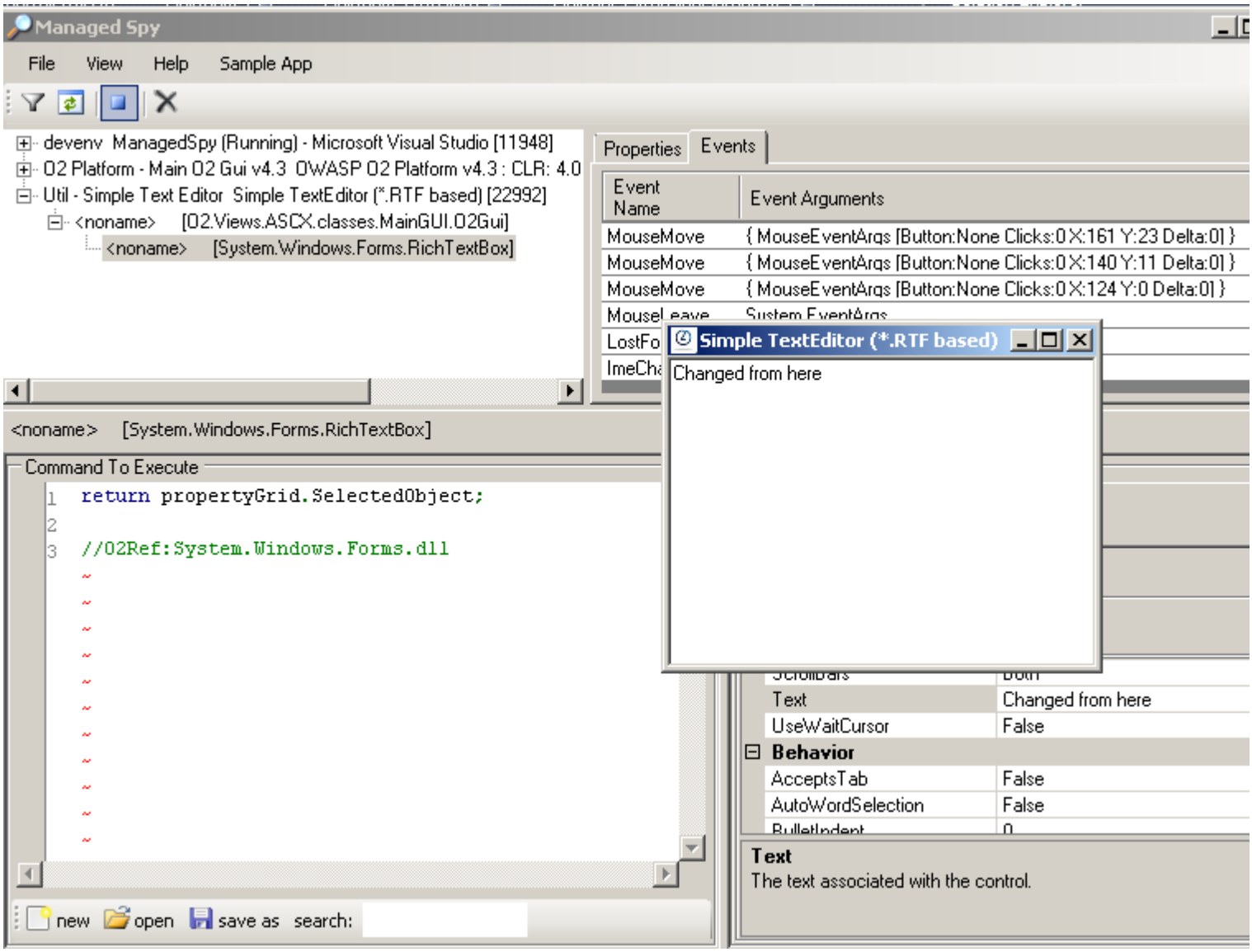
When executed there a couple little things to fix on the Gui (and it looks like the events got lost in the copy and paste



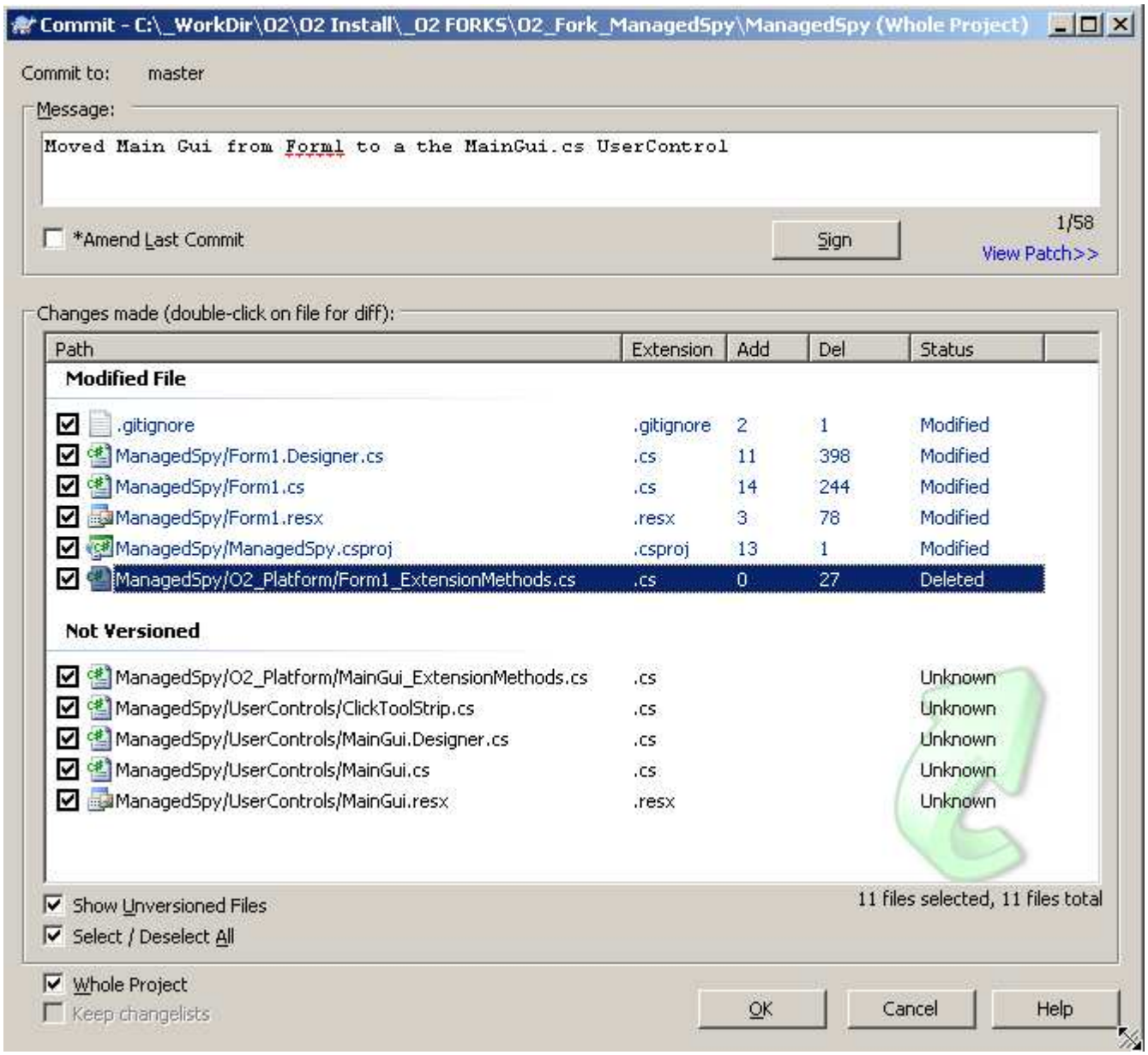
back in VisualStudio we can see the same probs in the Form Designer:



After fixing the GUI and adding the events the original *ManagedSpy* is back:

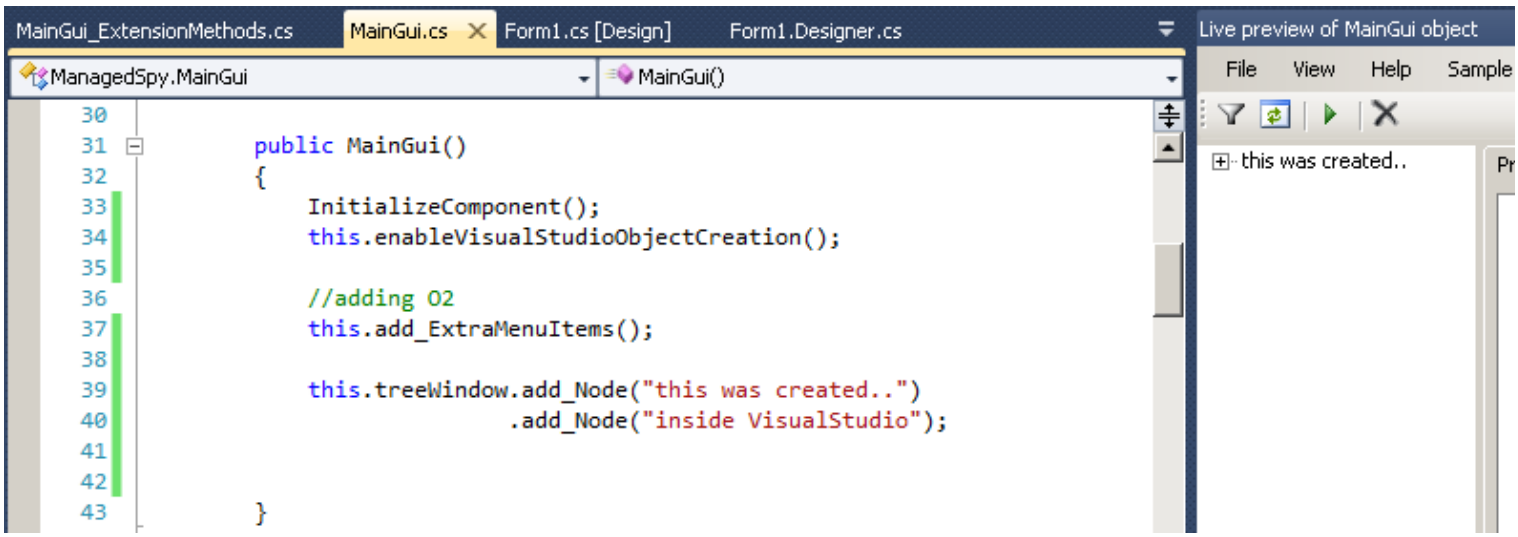


Commit the changes

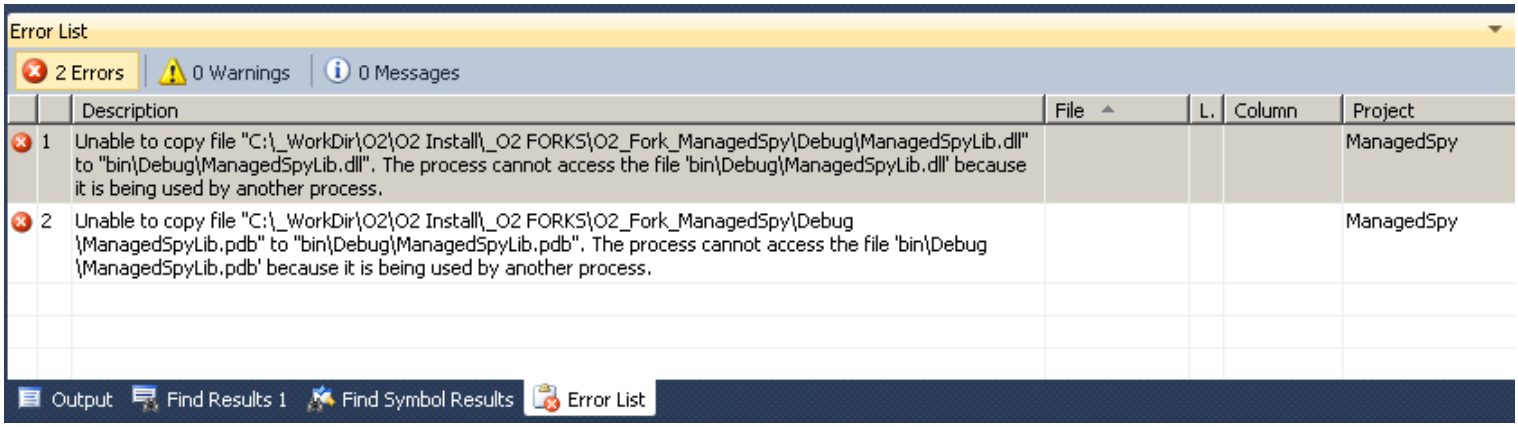


Live Preview of control created:

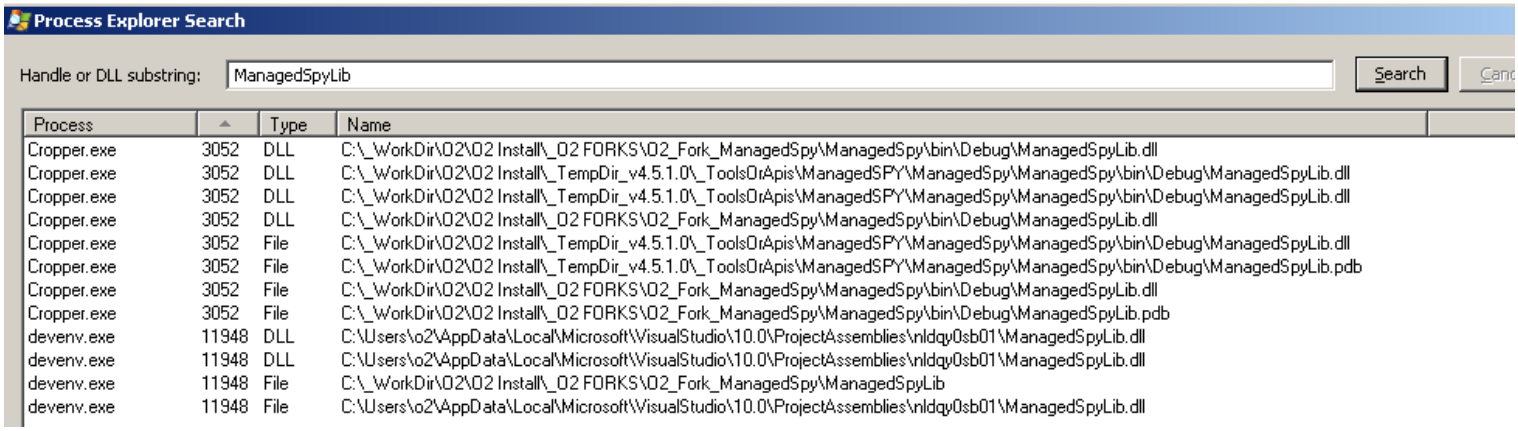
See document *ManagedSpy - Using O2 VS Extension to create ManagedSpy UserControls or Forms inside VisualStudio.pdf* for details on how the environment below was created that allows the live preview of the MainGui control inside VisualStudio



Note if you get this error:



is because this DLL has been loaded into a current .Net process:



In this case after stopping the Cropper.exe process, the compilation succeeds