Limnor Performer Import Wizard User's Manual

1 Introduction

Limnor Performer Import Wizard is a tool to convert ActiveX controls and .Net classes into Limnor Performers. Thus you can use those software components in your Limnor applications.

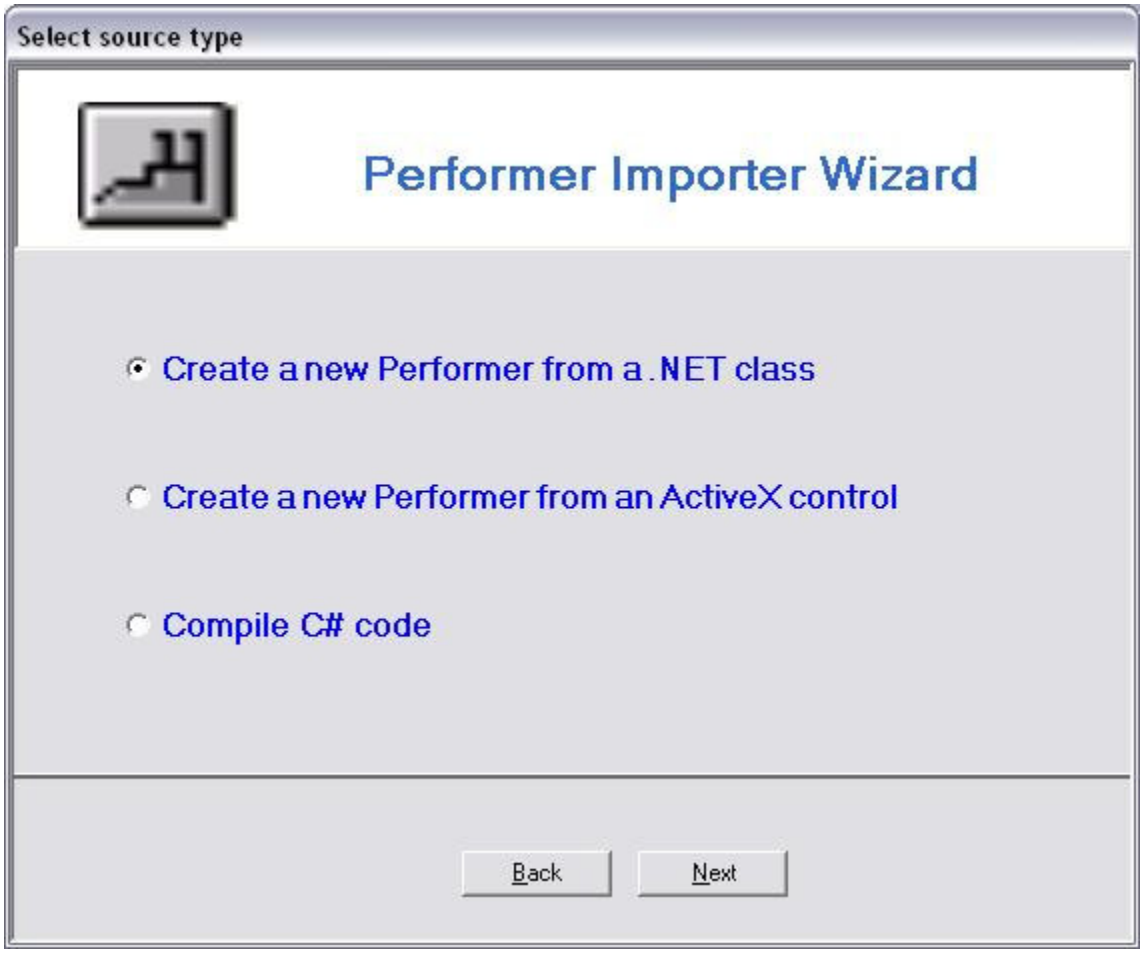
The properties, methods and events of an ActiveX control or a .NET class become the properties, methods and events of the Limnor Performer generated. The Wizard lets you choose which properties, methods and events to be generated.

It should be pointed out that codeless programming as the new generation of software engineering it is fundamentally different than current software engineering based on coding. ActiveX controls and .NET classes are designed for coding. Therefore the designs of ActiveX controls and .NET classes are, in most cases, not best for codeless programming. This Wizard just does a direct converting and cannot re-design them for best codeless usages. The Wizard will produce source code of the Limnor Performer generated, and you may hand-craft on the source code to create Performers best fit for codeless programming and your needs.

2 Choose Source Type

Click "Start" button to start the Wizard:

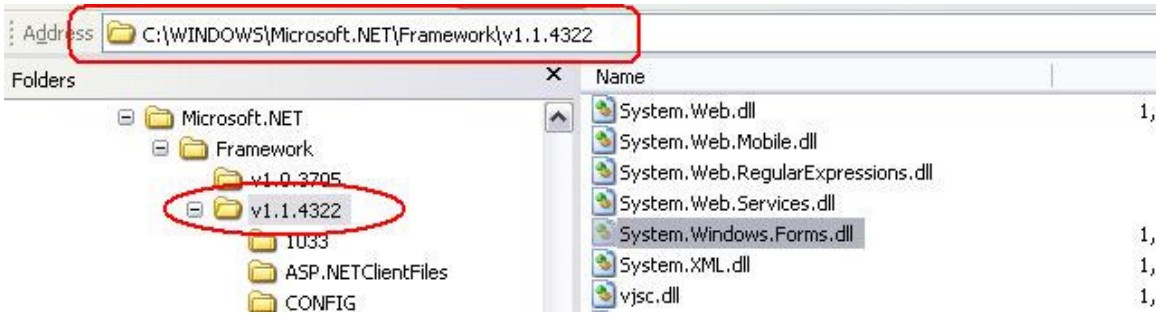
It allows you to choose one of 3 source types to start:



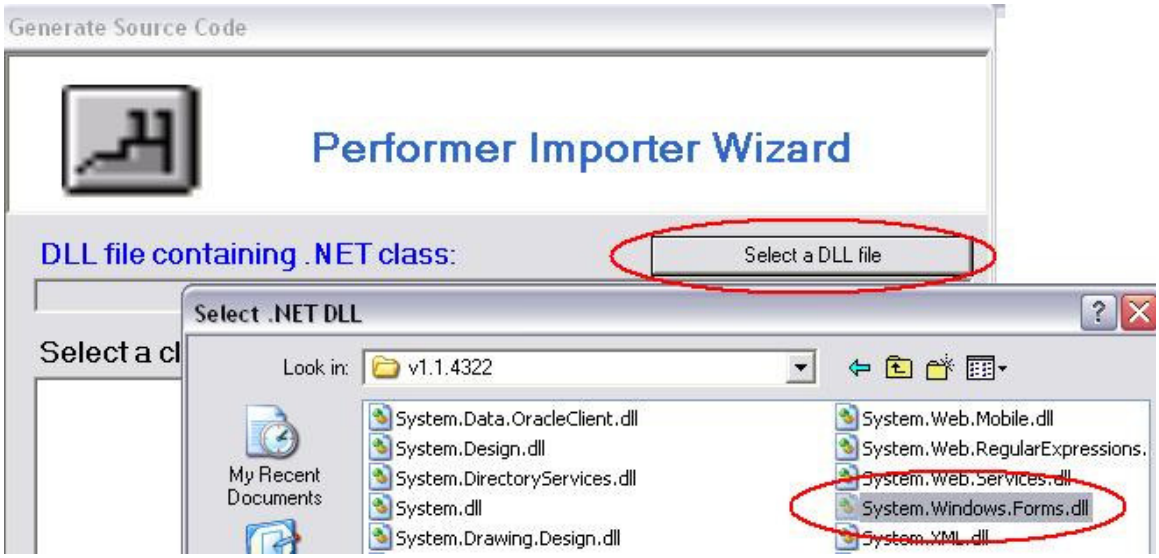
- .Net Class. This option lets you select a .NET DLL and select a class from the DLL to be converted into a Limnor Performer.
- ActiveX control. This option lets you select an ActiveX control. The wizard will convert it to a Limnor Performer.
- Compile C# code. This option lets you compile a C# source code file into a DLL file. You may modify the source code generated from the other options and use this option to compile your code into DLL files.

3 Convert .NET Classes

When you select “Create a new Performer from a .NET class” and click Next, it allows you to specify a DLL file containing the class you want to convert. You can select a DLL file made by you or some one else. Let’s use a DLL from Microsoft .NET Framework to do the demo. You can find all DLL for Microsoft .NET Framework in a folder named “Microsoft.Net\Framework\v<version number>” under Windows folder:

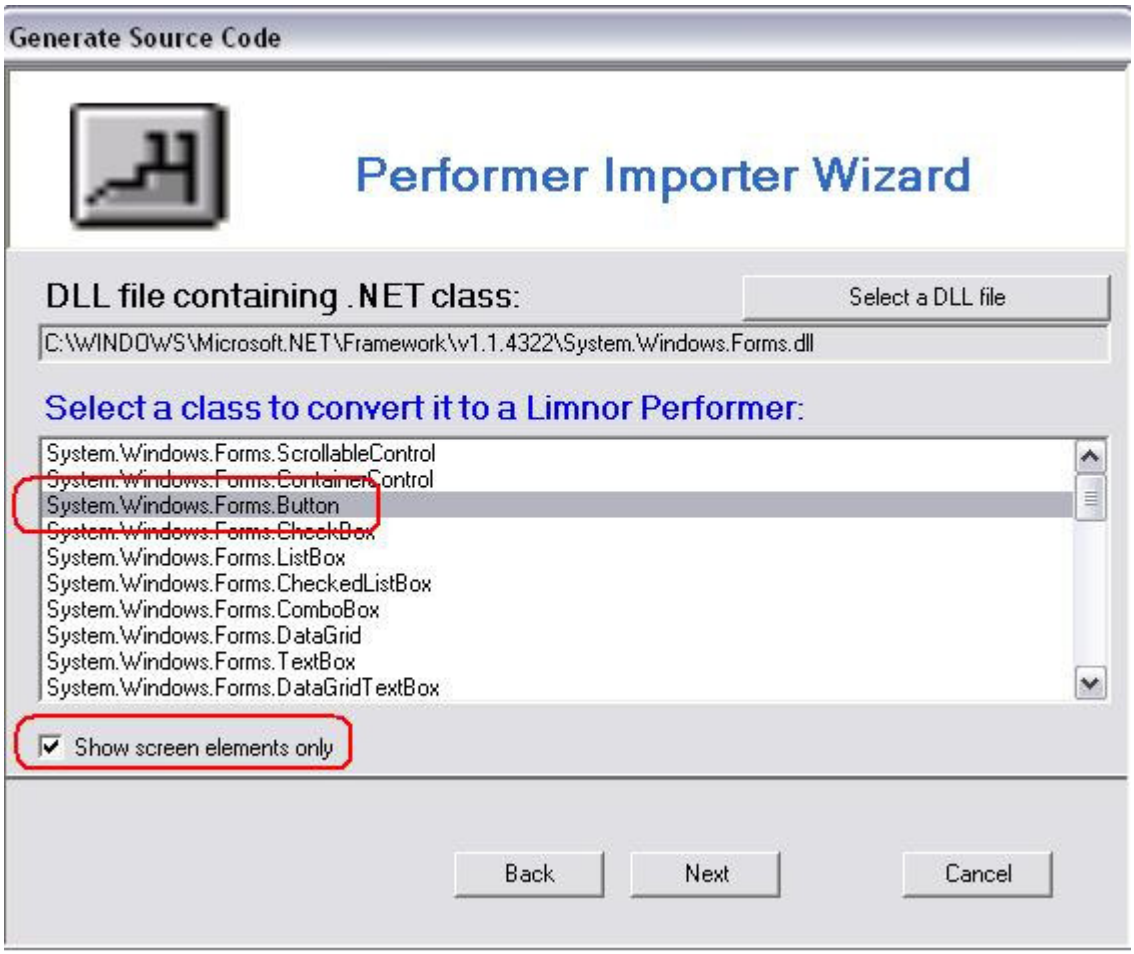


Suppose we want to convert the Button class into a Limnor Performer. The Button class is in System.Windows.Forms.DLL. Click button “Select a DLL file”, select System.Windows.Forms.DLL:

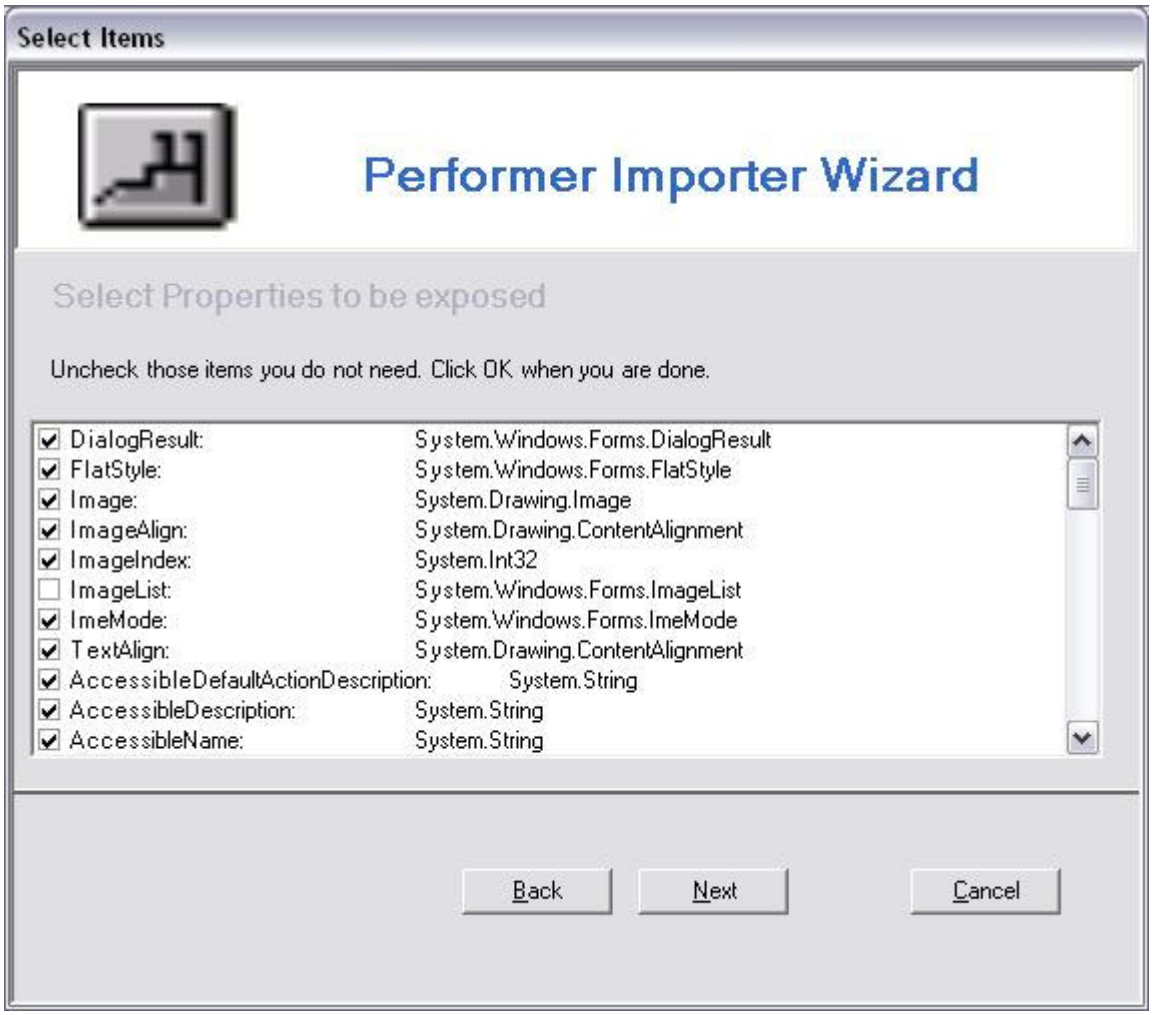


All public classes contained in the DLL file will be listed for your choice. There can be too many classes and not easy to find the one you want. If the class to be converted is a screen element (derived from Control class) then you may check the option “Show screen elements only”. With this option checked, non-screen-elements will be excluded from the list, making it easier to locate the class you want.

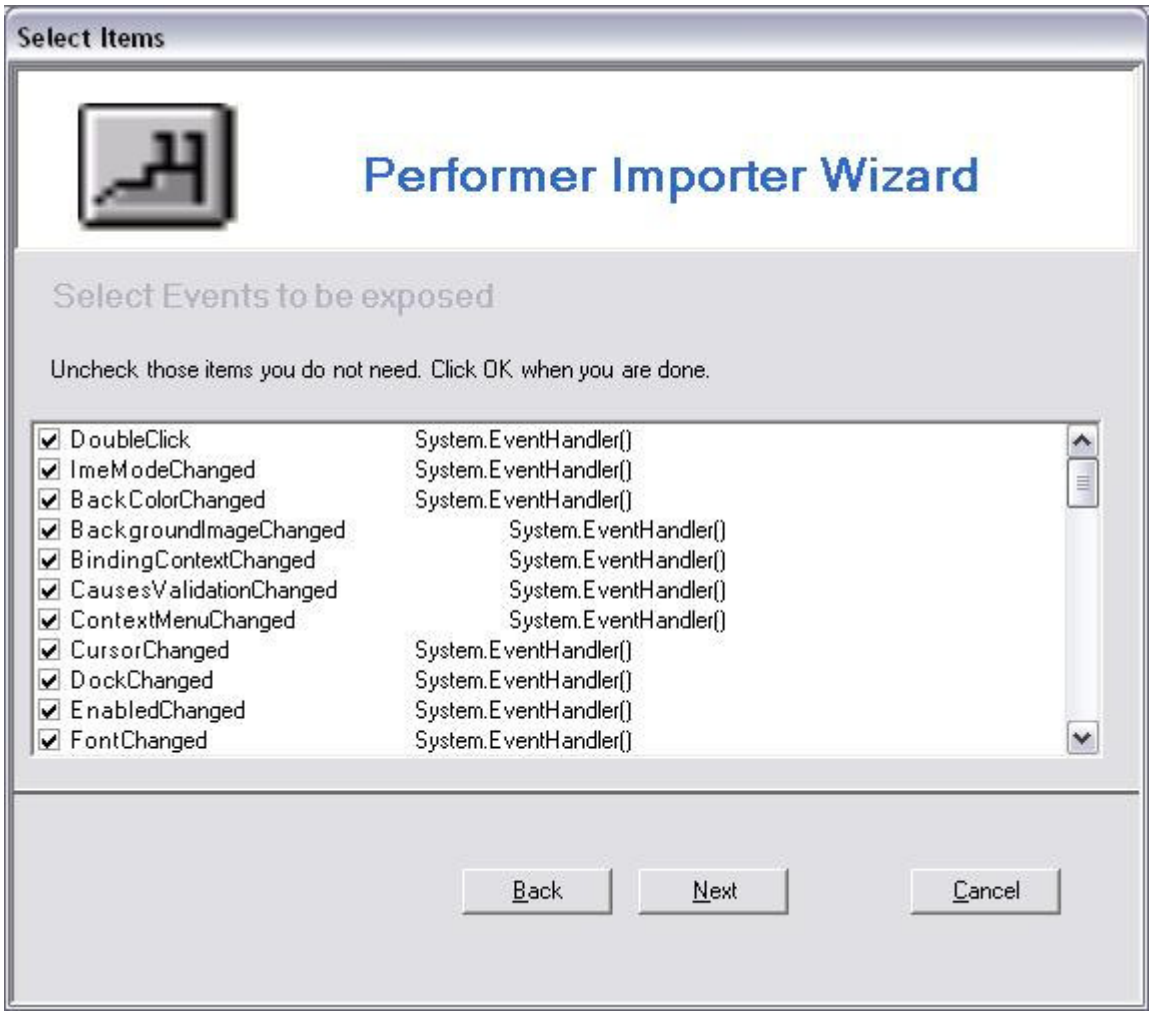
In our case, we want to locate and select the Button class, and click Next:



All properties of the selected class will be listed for you to make selections. Selected properties will become the properties for the new Limnor Performer:

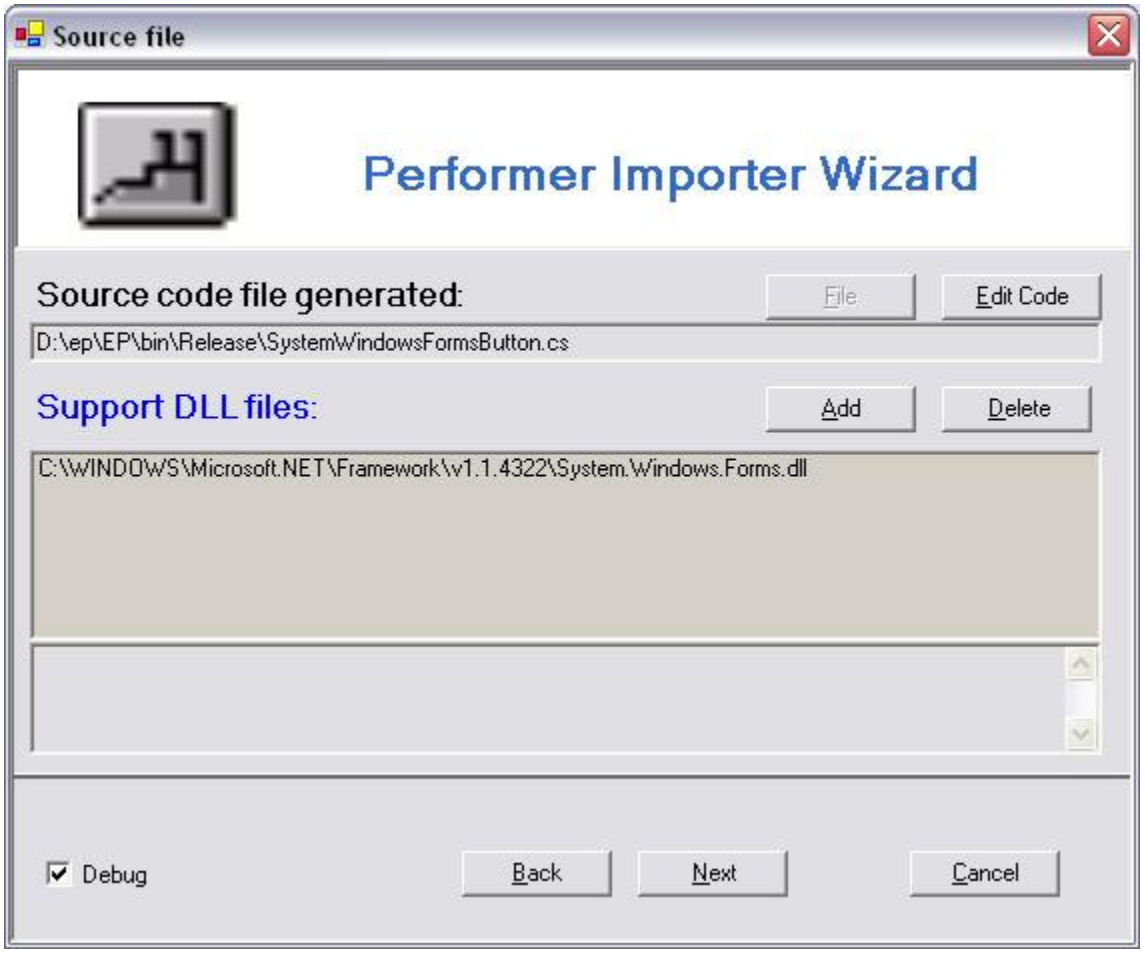


All events of the selected class will be listed for you to make selections. Selected events will become the events for the new Limnor Performer:



All methods of the selected class will be listed for you to make selections. Selected methods will become the methods for the new Limnor Performer:

After selecting properties, events and methods, click Next. A C# source code file is generated. The source code is for the new Limnor Performer:

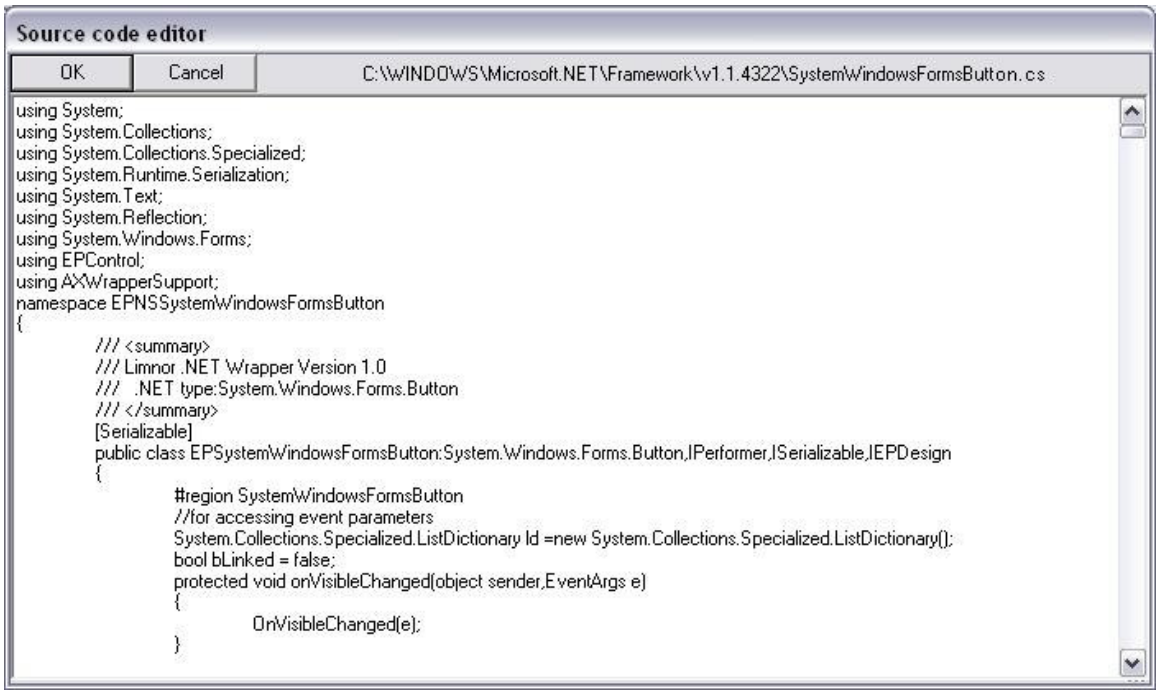


Clicking Next will compile the source code into a DLL. Once it is in DLL your Limnor applications can use it.

Before compiling it, if you know more DLL files are needed for the compilation, you may use “Add” button to add such DLL files into “Support DLL files” list.

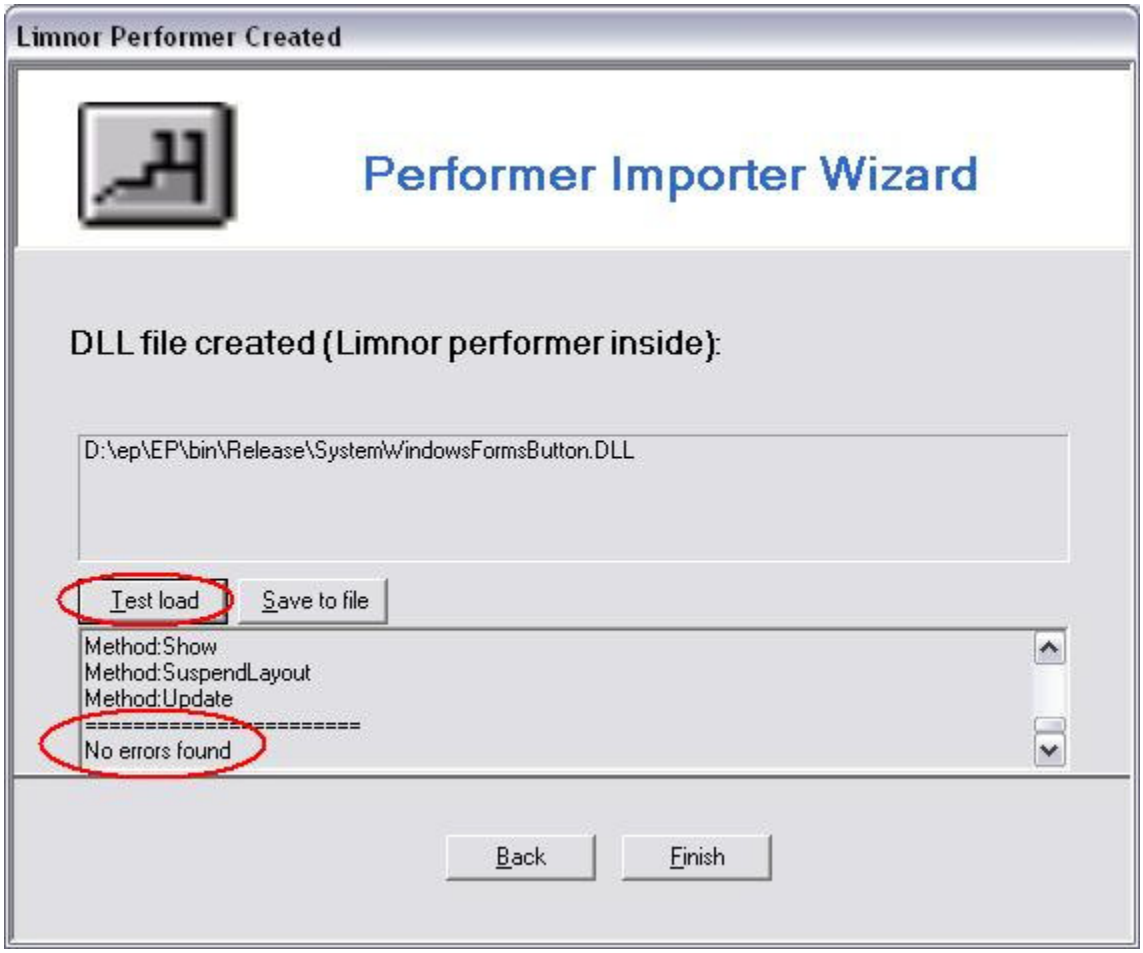
To generate debug information during compilation, check the box “Debug”.

You may modify the source code before compilations. Click button “Edit”, the source code is displayed for editing:



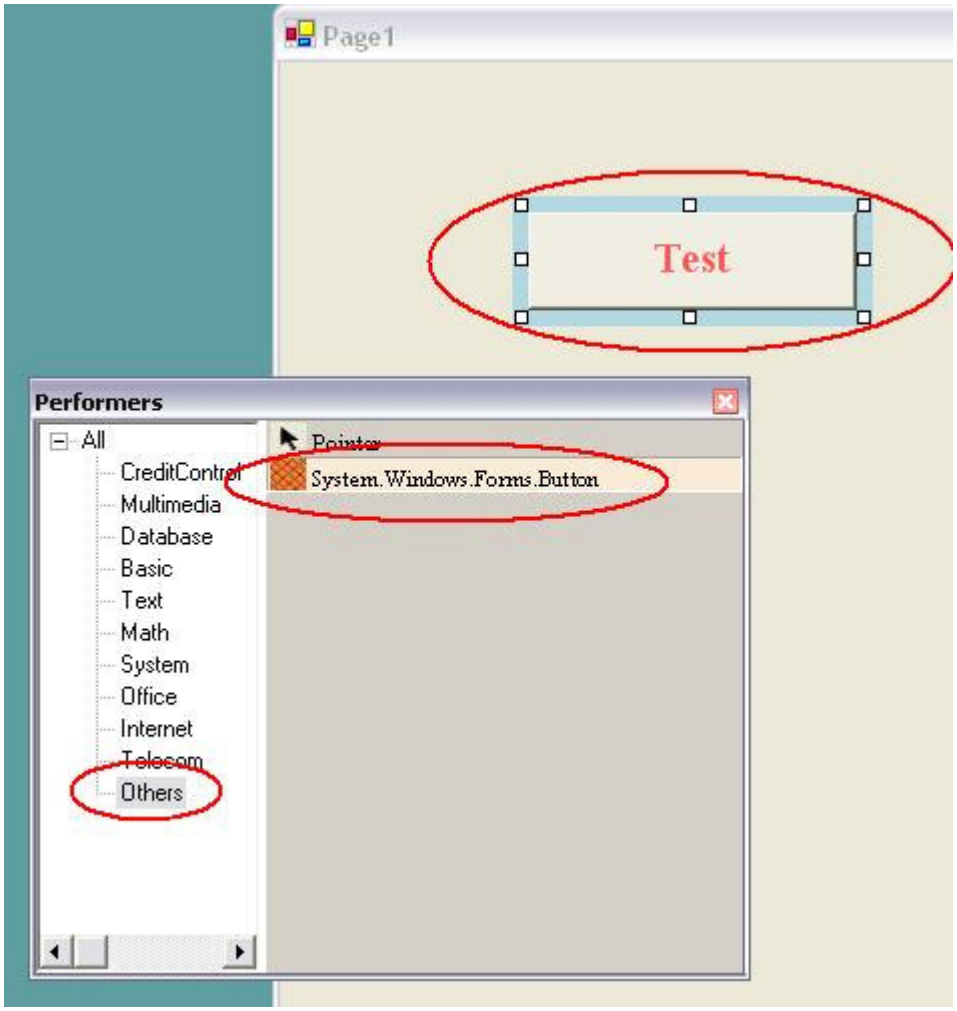
As you can see, the source code editor is a simple one. It does not have color coding as a professional source code editor. And it does not have intelli-sense as Visual Studio does.

When you are ready to compile the source code, click Next. If there are compilation errors then those errors will be displayed. You may edit the code according to the error messages, and re-do the compiling by clicking the Next button again.
If compilation succeeds then you may click “Test load” button to test the newly compiled DLL:



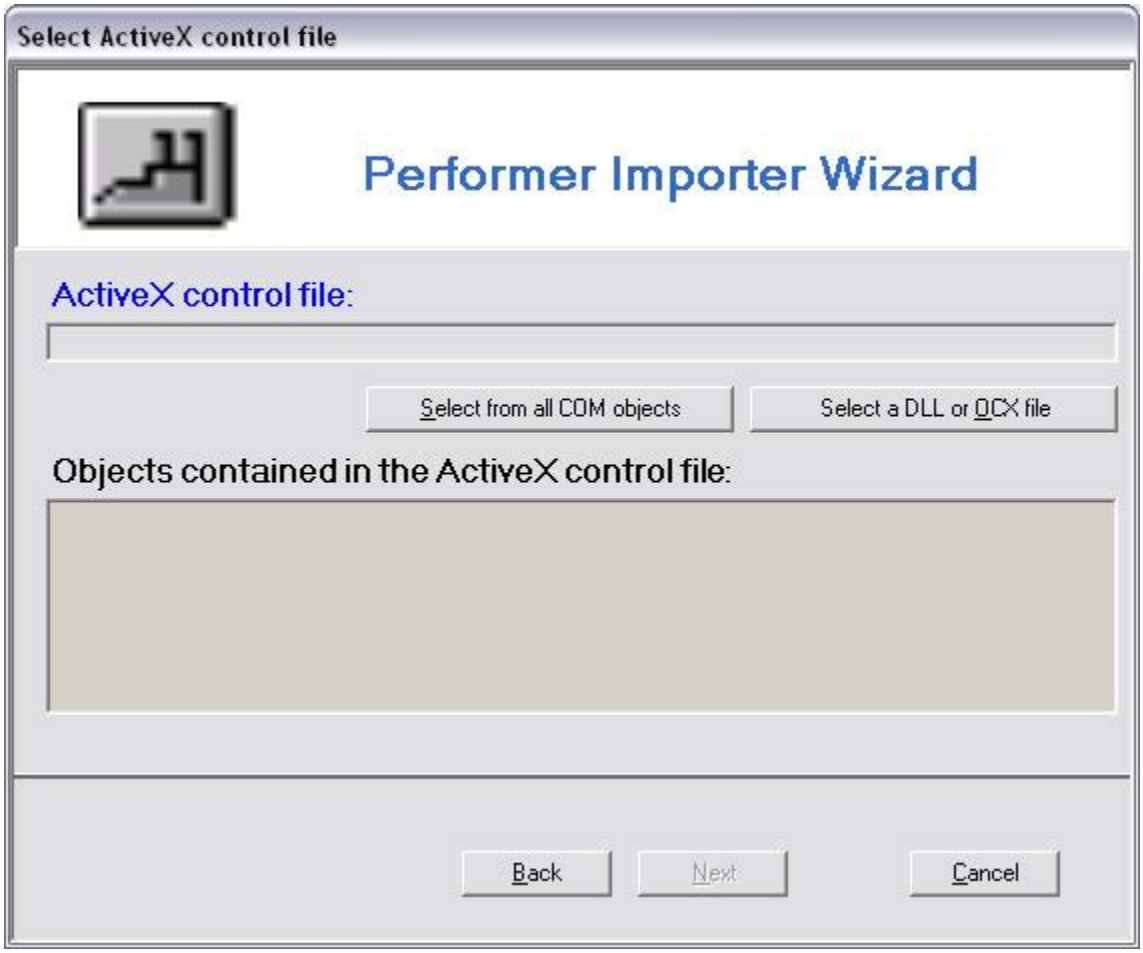
If the testing does not find any problems then you can see the last line of the message is "No errors found". We are done.
If the testing finds problems then you may click "Save to file" button to save the test results to a text file for later references. You may modify the source code according to the test results and recompile it. Before you can re-compile the source code, you need to close this wizard program and re-run it because the testing will cause the operating system to lock the DLL file.

With the new DLL compiled and testing being OK, we will see the new Limnor Performer in the toolbox under "Others" category:

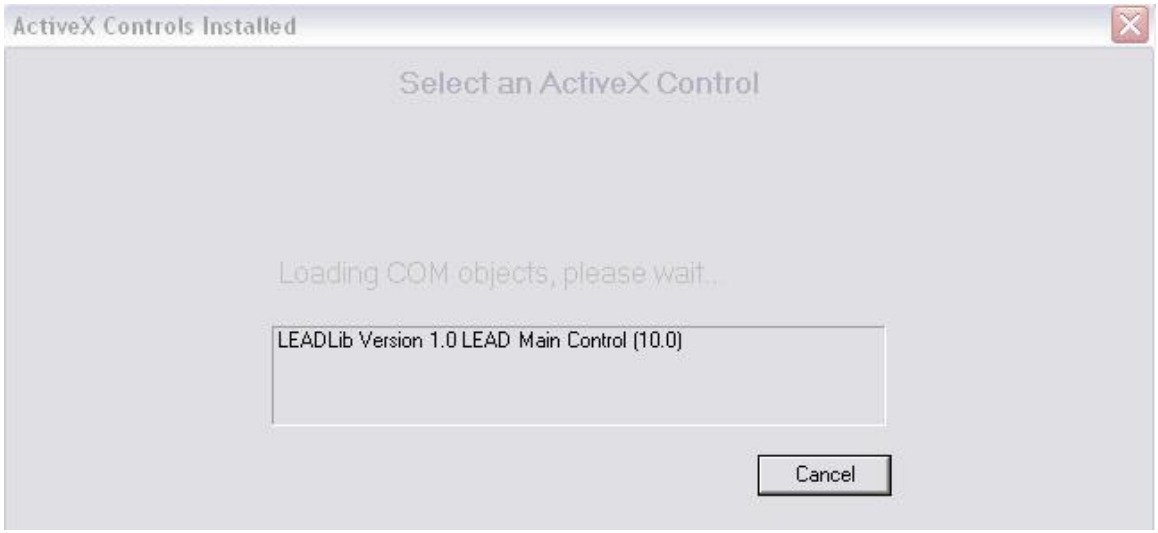


4 Convert ActiveX Controls

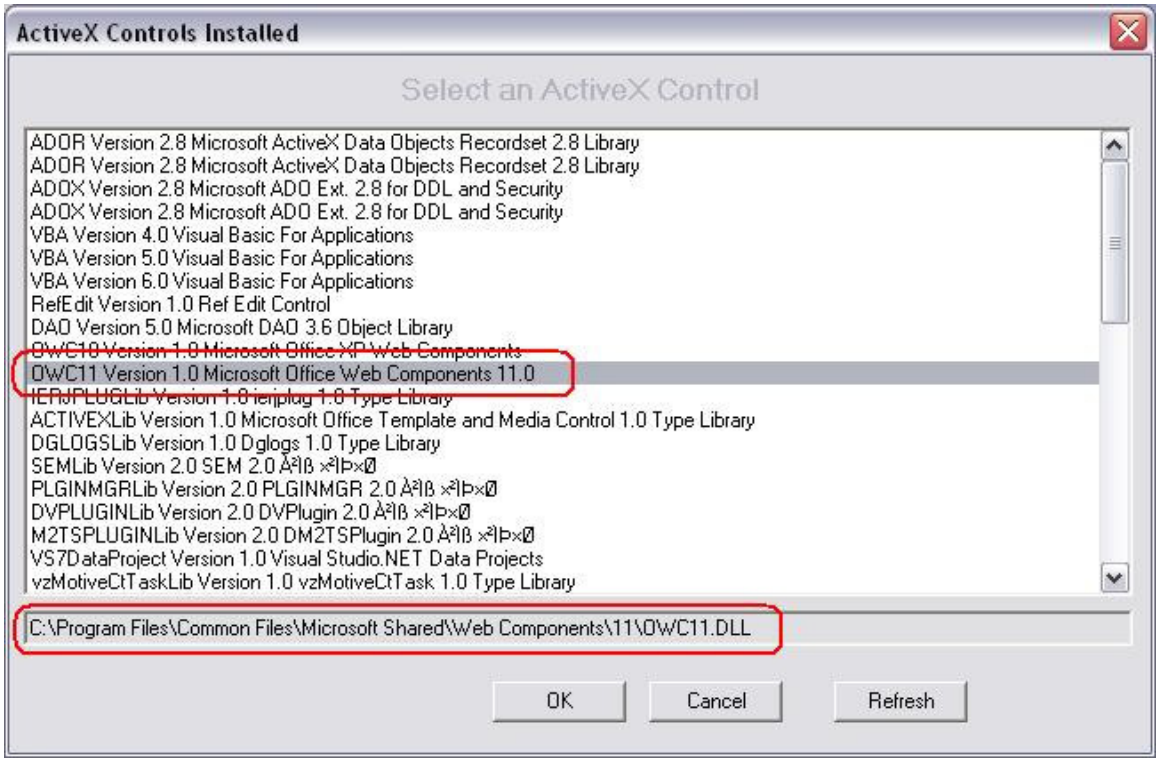
When you select “Create a new Performer from an ActiveX control” and click Next, it allows you to search for ActiveX controls:



Click button “Select from all COM objects”. It will start searching your computer for all COM objects:



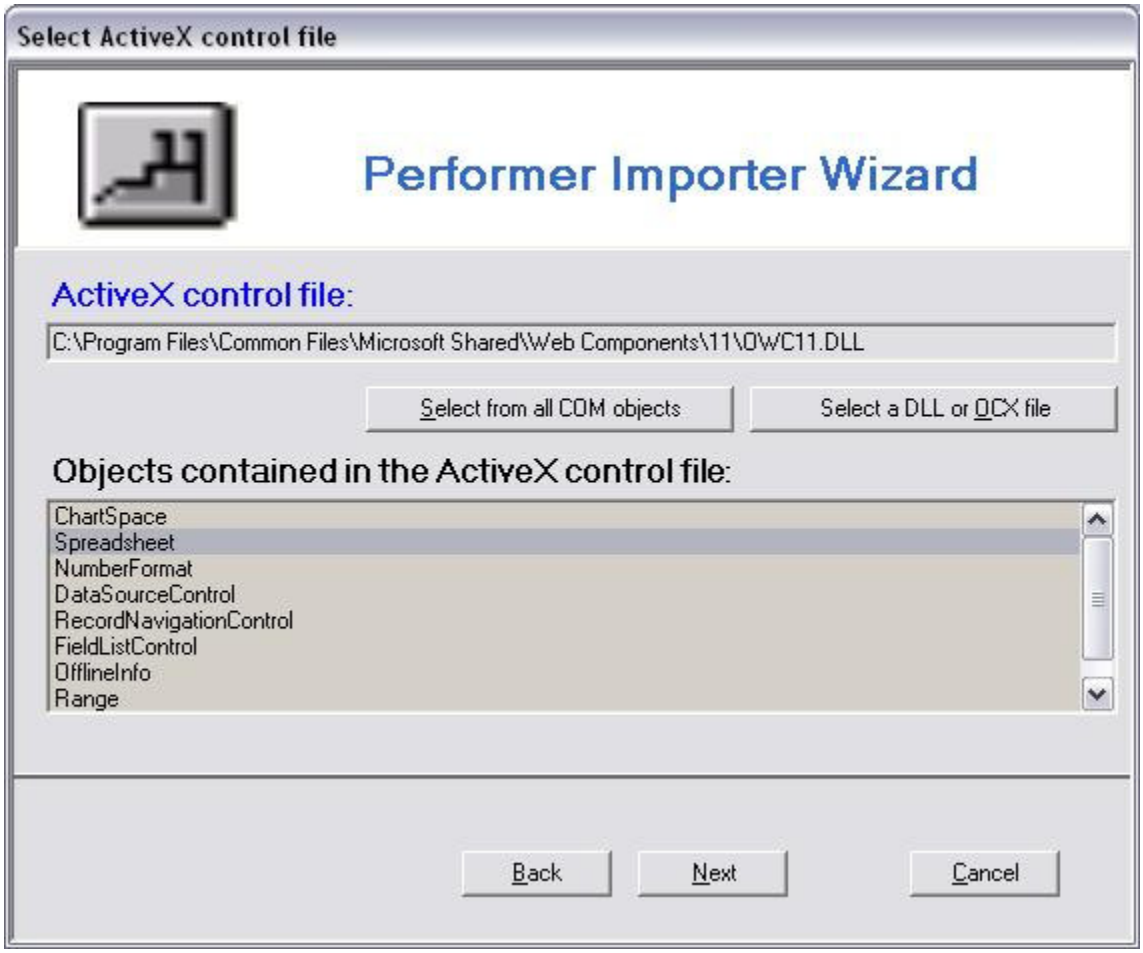
You may then select the ActiveX control you want from the list:



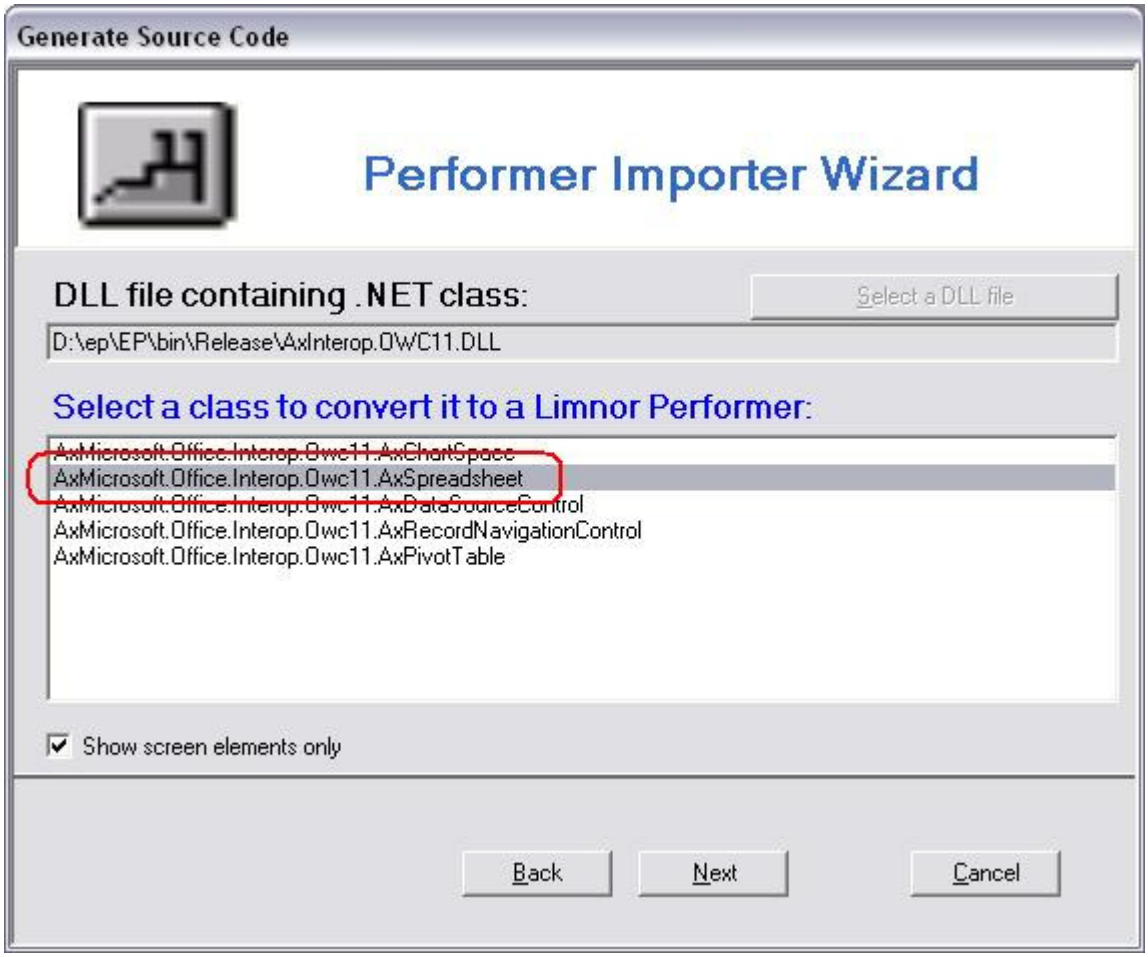
You can see the DLL file for the selected ActiveX control is also displayed.

If you know the DLL/OCX for the ActiveX control then you do not need to search for all COM objects. You may click button “Select a DLL or OCX file” to pick that file.

When you select a COM file, all COM objects contained in the file will be displayed. Make sure the ActiveX control you want to convert is in the list.

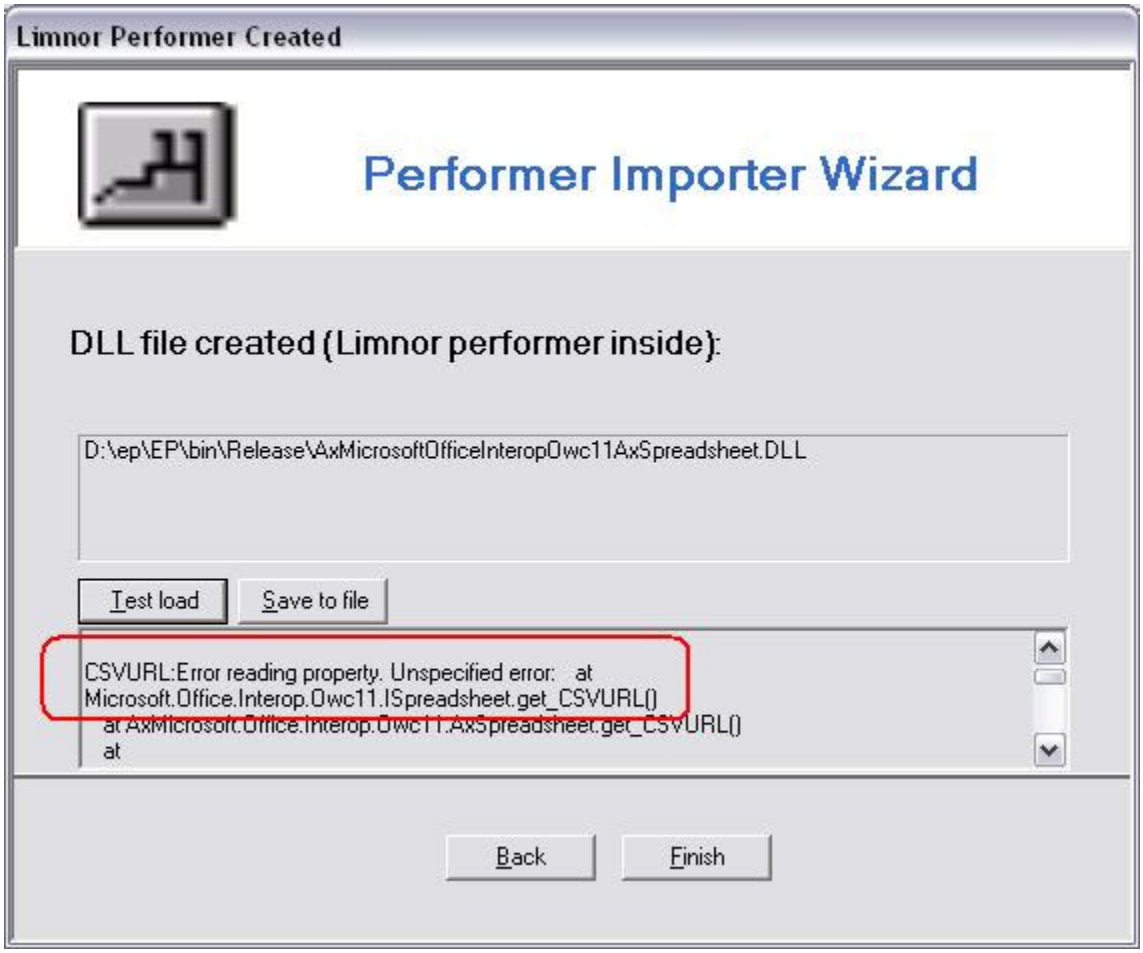


Click Next. A .NET DLL file will be created. It wraps all COM objects to .Net classes:



At this stage it is the same as we have selected option of “Create a new Performer from a .NET class”. The .NET DLL file we used is the newly created DLL file for wrapping the ActiveX control. In our example here, it is AxInterop.OWC11.DLL. The only difference from the option of “Create a new Performer from a .NET class” is that the button “Select a DLL file” is disabled, because we do not need to select a .NET DLL.

The next operations are exactly the same as we described for option of “Create a new Performer from a .NET class”. Only that for ActiveX controls, you may see some properties cannot be used. When we get through all the steps and click “Test load” button, it will show the properties failed to load:

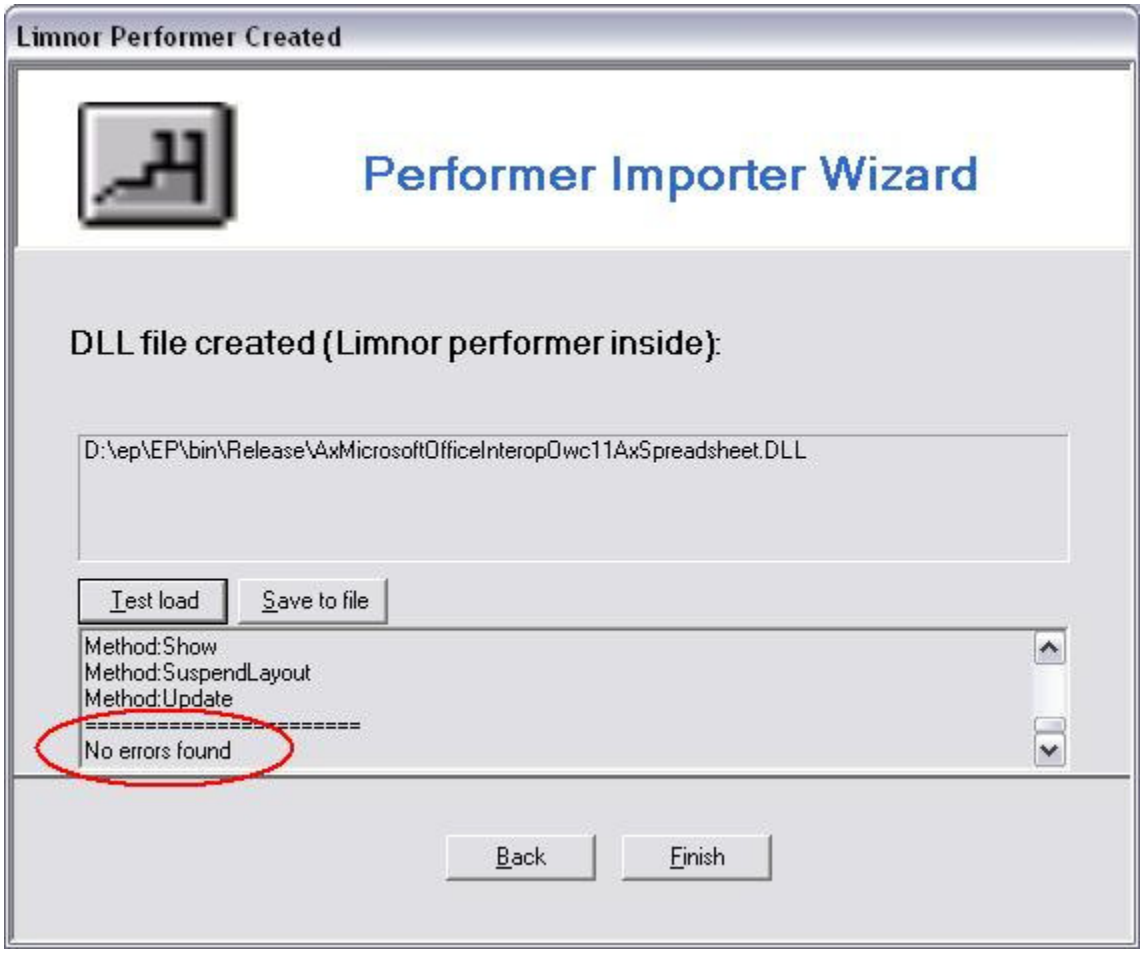


You may click “Save to file” button to save the results to a text file for references. From the text file we can see such errors:

```
CSVURL:Error reading property. Unspecified error:
HTMLURL:Error reading property. Unspecified error:
XMLURL:Error reading property. Unspecified error:
```

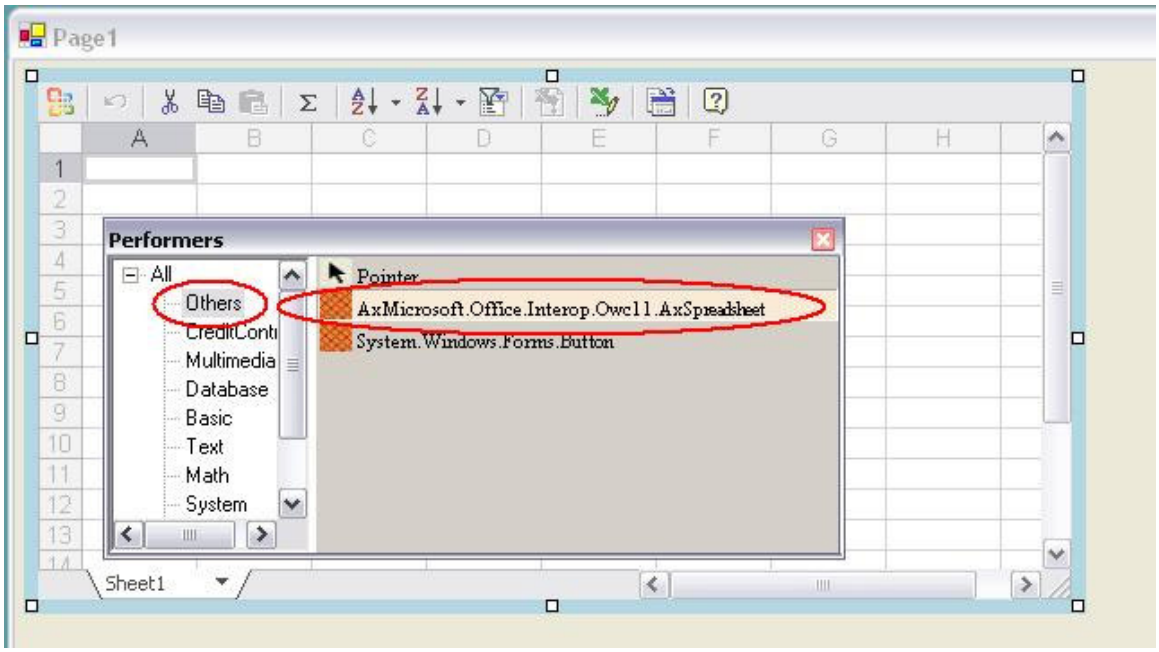
We need to close this wizard program to release the DLL we generated. We re-start the wizard program, and re-do it. At the step of selecting properties, we may uncheck those properties which cause problems:

Now this time no errors are found:



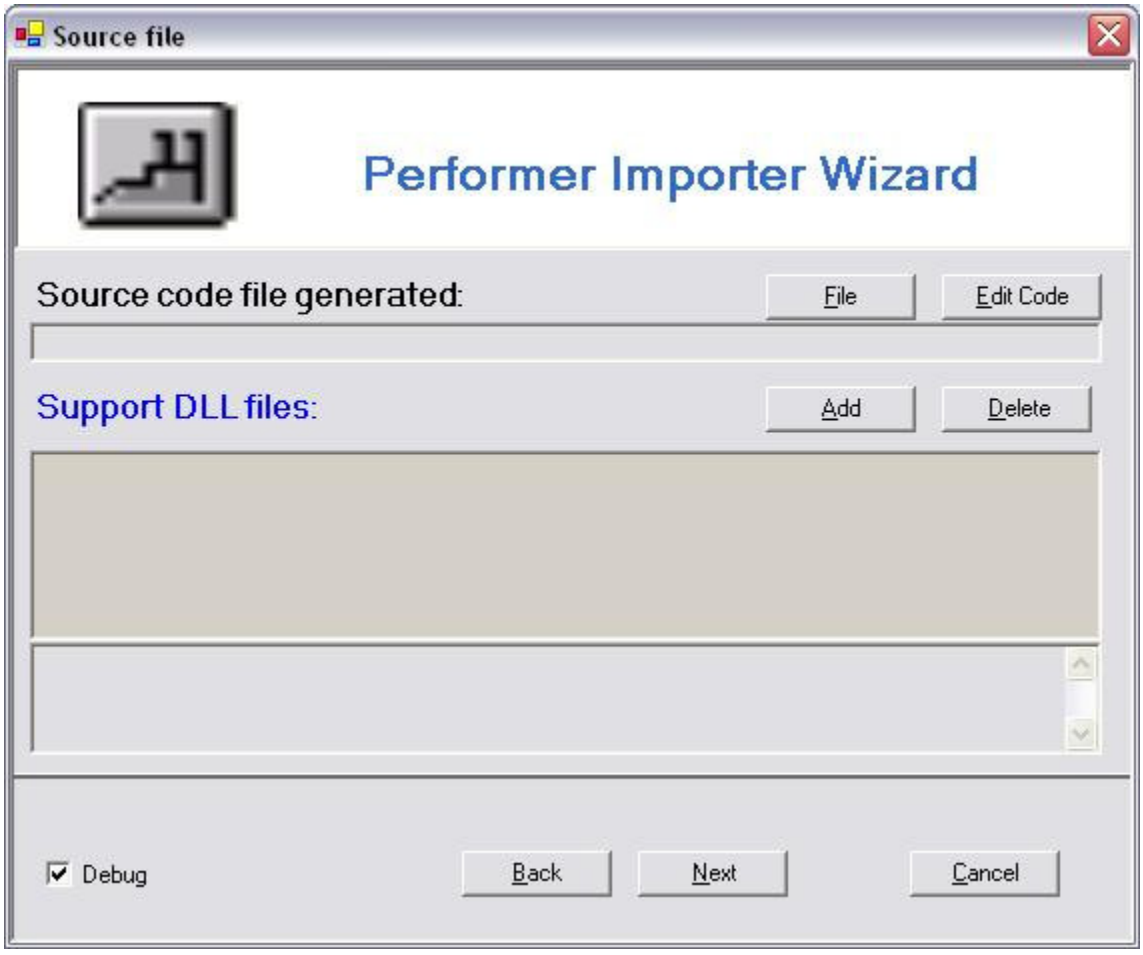
When you are doing a converting and you do need those properties causing errors, please contact us at support@limnor.com. We may hand-craft it to put those properties back.

Now we may start using the new Performer in our applications:



5 Compile C# code

When you choose option “Compile C# code”, the Wizard jumps to the step of before compiling and enables the “File” button:



Click “File” button to specify the source code file to be compiled. If your source code uses classes from other .NET DLLs then you need to add the DLL files to the “Support DLL files” list:

Click Next. The source code will be compiled into a DLL file.

6 Support and Feedback

For questions and feedback, please contact support@limnor.com

