TYX CORPORATION

Productivity Enhancement Systems

Creating an I/O Subsystem Resource in .NET

Prerequisites:

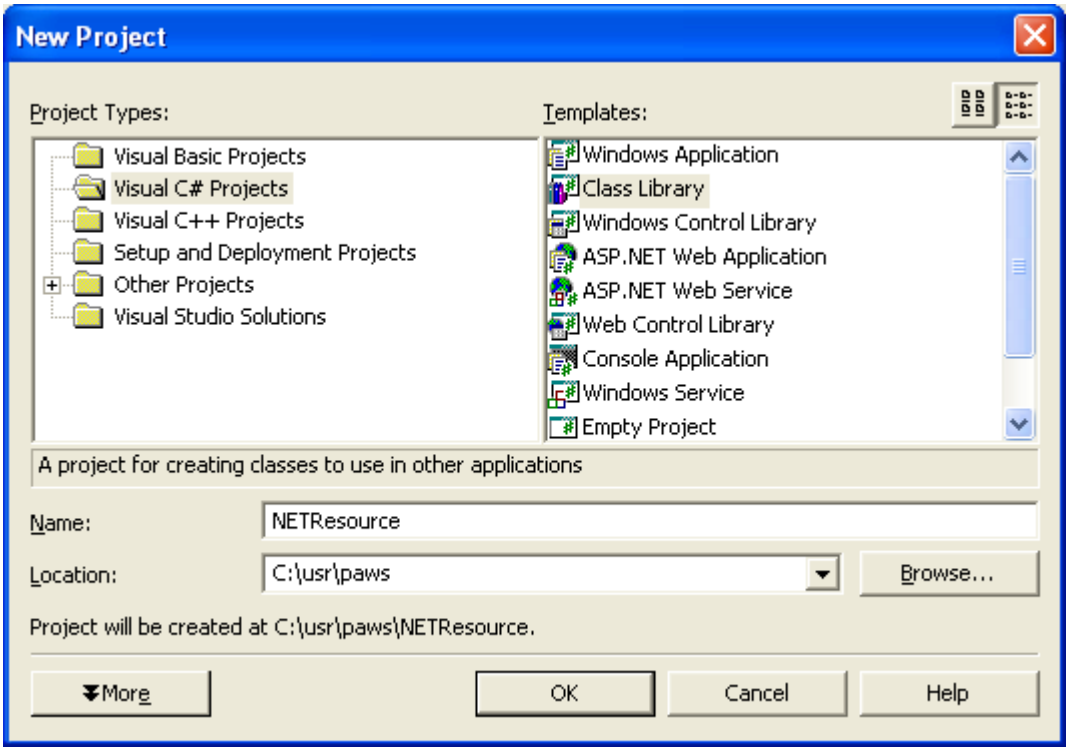
Microsoft's .NET Framework installed on the system.
Paws Studio & RTS version 1.34.7 installed on the system.
Microsoft's Visual Studio IDE 7.0 or higher (not a necessity).

Task:

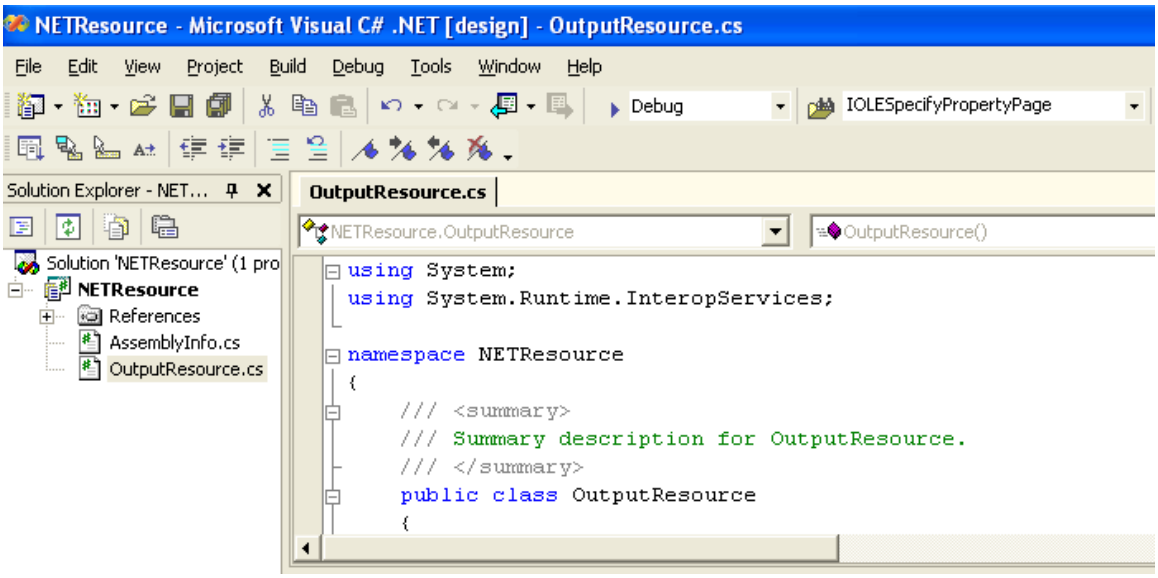
To create a basic "**OUTPUT**" resource in the .NET environment and configure the RTS to use it. The resource will display a message box window when it's "**Open**" and "**Close**" methods are called. Just before the resource closes it will display a message box window containing a concatenation of all the "**Output**" calls that this resource received (this would be configurable). The resource will also have its own property page, containing a check box which would control viewing of the concatenated RTS "Output" string. The settings of the resource will be serialized and will be available for another RTS session.

Resource Component:

- Using Visual Studio IDE create a new C# "**Class Library**" project

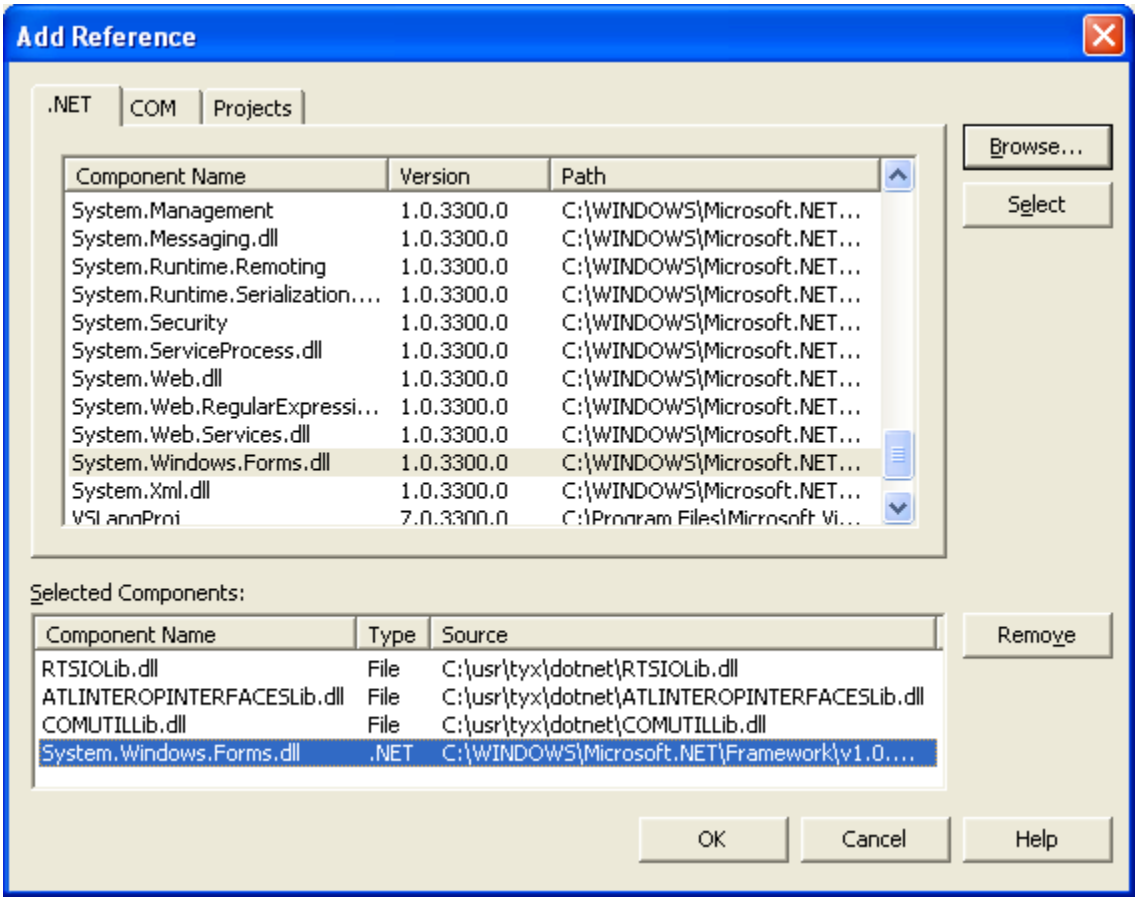


- Update the name of the C# file to "**OutputResource.cs**", class name to "**OutputResource**" & the namespace to "**NETResource**".



- Add the following references to the project.

Select Projects Tab on the "**Add Reference**" dialog and browse to the <usr>\tyx\dotnet folder.

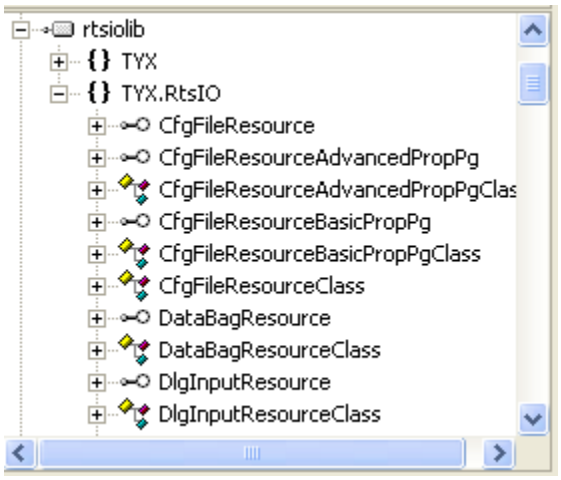


Select the RTSIOLib.dll, ATLINTEROPINTERFACESLib.dll & COMUTILLib.dll RCW's and click OK. Also, under the .NET tab add System.Windows.Forms.dll.

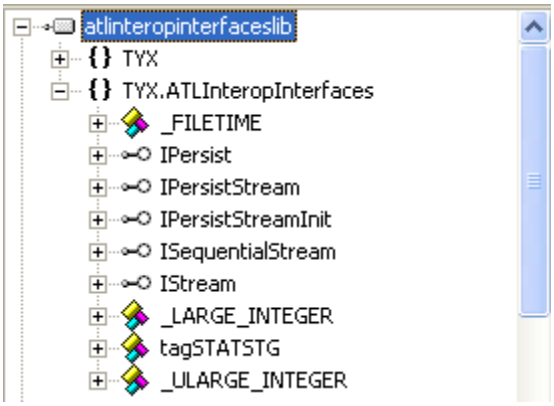
[**Note:**

This Microsoft supplied component will be required only if the resource needs to display a GUI. In this example we will display a Message Box so the reference has been added].

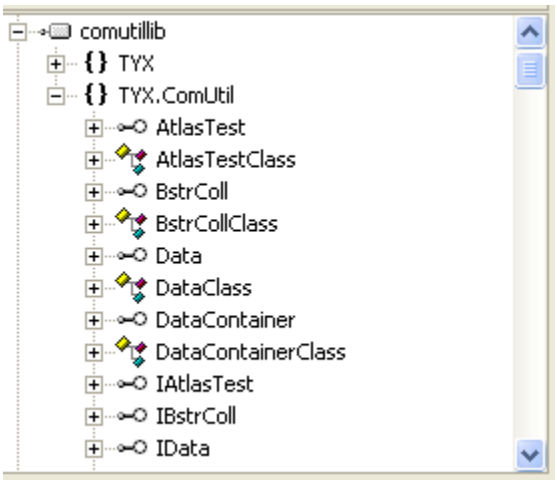
RTSIOLib.dll is the RCW containing the definition of the "**TYX.IIOResource**" & "**TYX.ITextResource**" interfaces which will be implemented by this OutputResource component. On encountering ATLAS "**Output**" statements the RTS will invoke methods on the "**TYX.IIOResource**" & "**TYX.ITextResource**" interfaces.



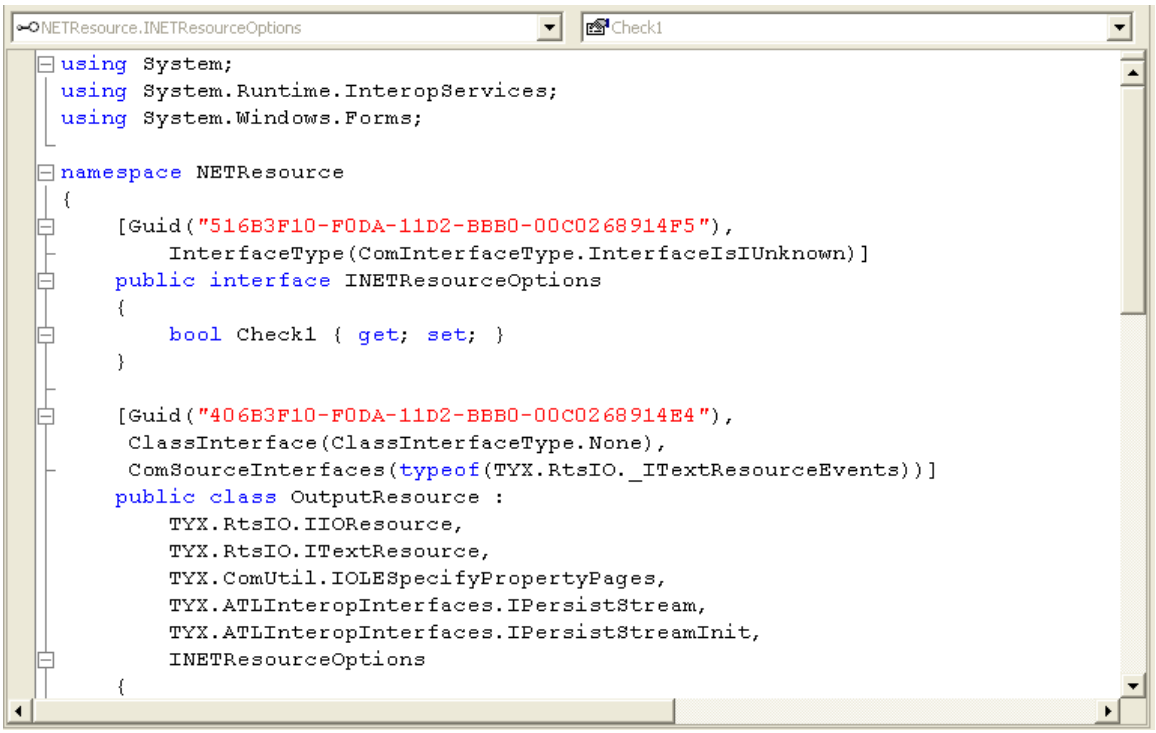
ATLINTEROPINTERFACESLib is the RCW containing the definition of the "**TYX.IPersistStream**" & "**TYX.IPersistStreamInit**" interfaces. These interfaces are implemented by the OutputResource component to provide serialization for the properties of the resource.
[**Note:**
If the component does not need to provide serialization of its settings the reference to ATLINTEROPINTERFACESLib RCW can be ignored].



COMUTILLib.dll is the RCW containing the definition of the "**TYX.IOLESpecifyPropertyPages**" interface. This interface is implemented by the OutputResource component to provide the "**PROGID**" of the component(s) that implement the property page(s) for this resource.
[**Note:**
If the component does not need to provide a property page the reference to COMUTILLib.dll RCW can be ignored].



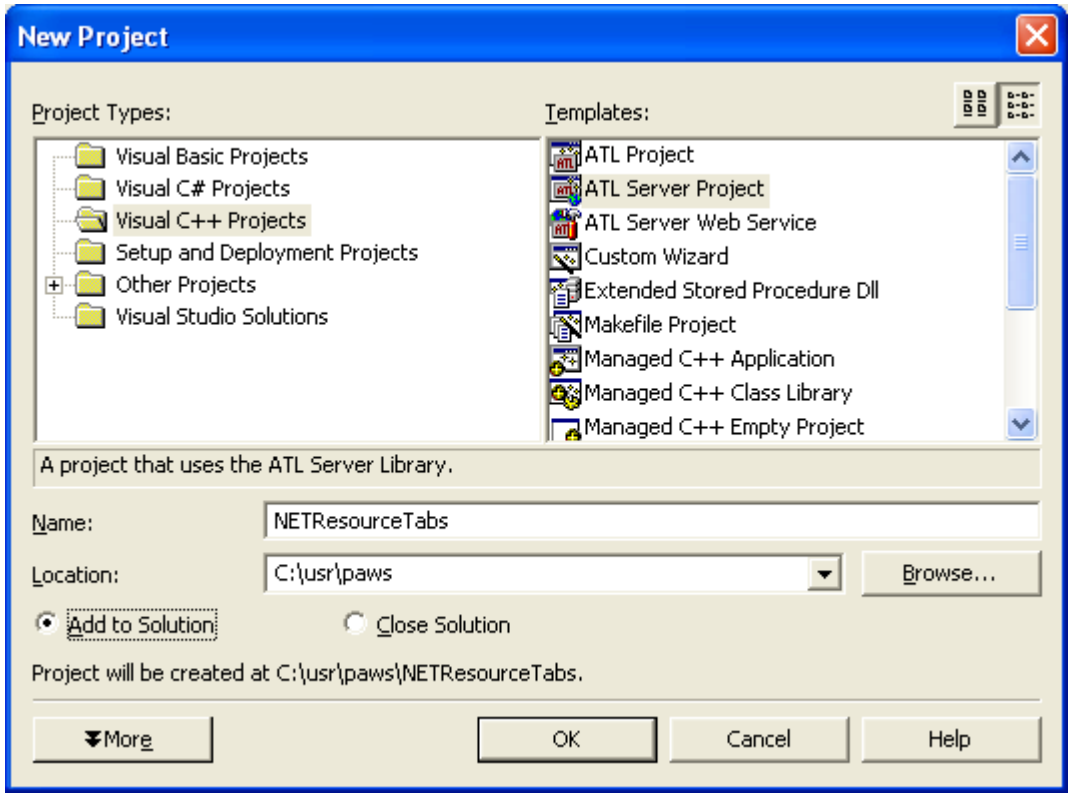
- Make the OutputResource component implement the interfaces as shown below



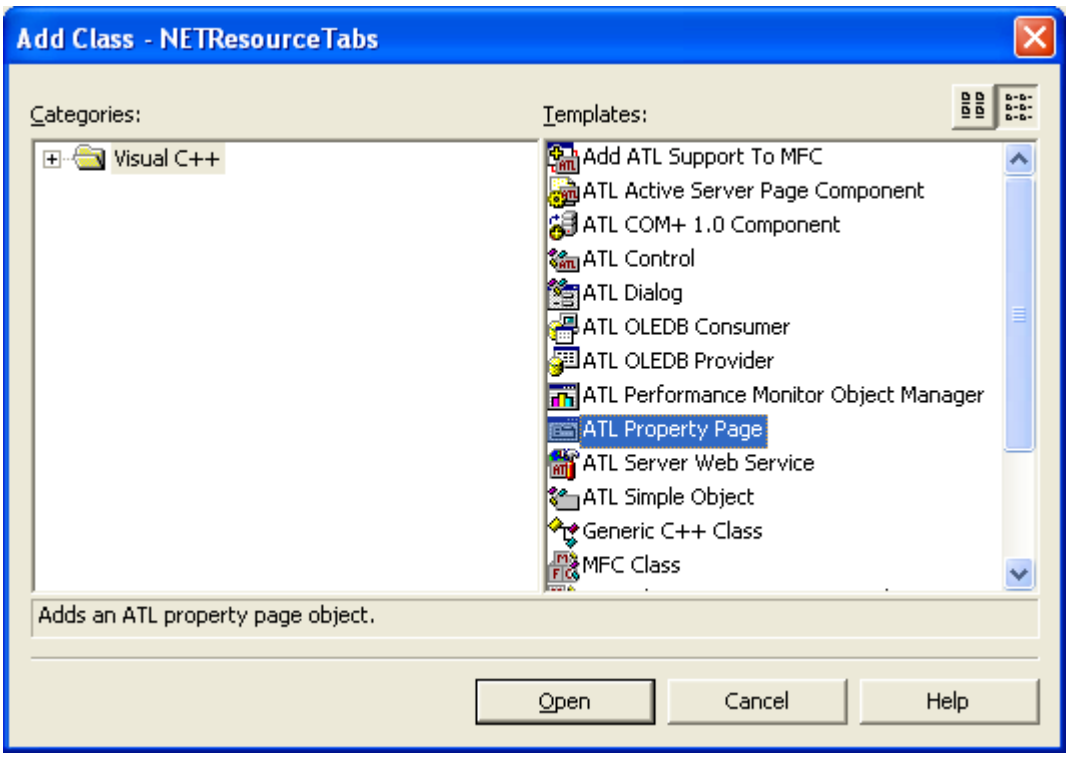
INETResourceOptions is an interface to control the update of properties of the OutputResource component which are displayed in the property page. For this example it contains a single Boolean property "**Check1**" which is updated via the "**Display Output Data**" checkbox on the property page.
[**Note:**
Please see "**NETResource Files.zip**" for interface implementation code].

Resource Property Page Component:

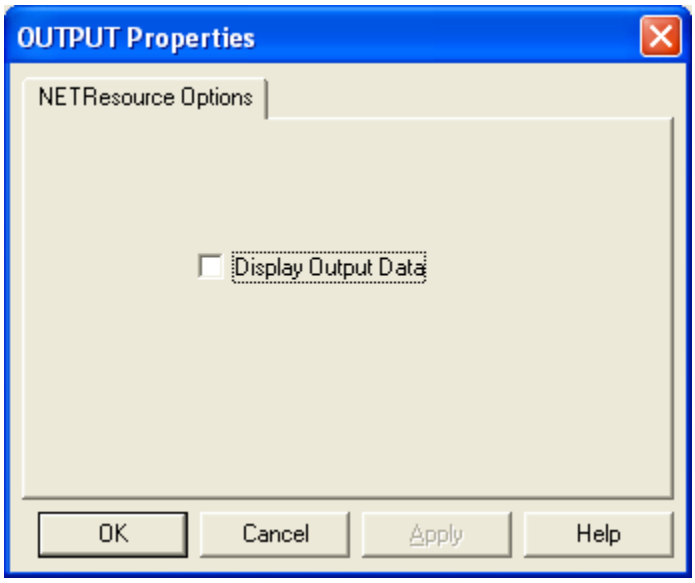
- Using Visual Studio IDE create a ATL Server Project



- Add a Property Page class to the project & name it "**Page**"



- Update the resource of the property page as follows



- We implement the Property page object by querying for the "**INETResourceOptions**" interface, which is implemented by the NETResource [C# resource] component and updating it as per user selections.

- NETResource component implements the "**IOLESpecifyPropertyPages**" interface and within the "**GetPages()**" method returns the **"NETResourceTabs.Page"** which is the "**PROGID**" of the Property Page component.

Persisting Property Page Settings:

- NETResource component implements the "**TYX.IPersistStream**" and "**TYX.IPersistStreamInit**" interfaces. Key methods of these interface like InitNew(), Load() & Save() allow the resource a chance to serialize its settings. In this example the "**Check1**" property of the "**INETResourceOptions**" interface is serialized.
  [**Note:**
  Please see "**NETResource Files.zip**" for interface implementation code
  The file currently provides references to Interop assemblies distributed with PAWS & RTS expecting this to be installed under "**C:\<usr>\tyx**" directory].

Configuring RTS & Testing the Resource:

- Verify that the new .NET assembly is correctly registered in the GAC. You could use the "**..\Miscellaneous\bin\register.bat**" as an example to register the assembly.
- Select "**Control**"→"**Options**" and click the "**RTS Property Pages**" button to view the RTS Settings. Switch to the "**IOSubsystem**" Tab and configure the "**OUTPUT**" resource to be "**NETResource.OutputResource**"
- You should be able to click "**Properties**" and configure the properties for the resource.
- Save the RTS configuration.
- Re-start RTS and load & run the test paws project located under "**..\Miscellaneous\PawsProject\PawsProject.PAW**". Verify that you can view the desired output message boxes.
  [**Note:**
  Open message displays when you open the project. Close message displays when you unload the project. The concatenated output will be displayed if the "**Check1**" property in the property pages for the "**OUTPUT**" resource has been checked].