SSIS-Web Service With Xml and Error Handling in control flow

"Web service task" let us execute the web service method and then store the result into a variable or inside a file.

This variable or file may act as an input for some other task.

Note:

- Intent of this article is to explain SSIS not web services. We are assuming that you have a basic understanding of web services. If you are new to web services then in step 1 and step 2 we have shared link which will make you familiar with web services.
- In real life scenario as a BI developer, mostly you are not required to worry about creation of web services. You will get metadata of a service in the form of WSDL from external sources (may be from some other developers or organization). Your task will be adding "Web Service Task" and that's what we will learn.

Step 1: Create a Demo WCF Service

Please go through the following link for getting help of creating a web service project.

Link: https://www.c-sharpcorner.com/article/create-wcf-web-service-in-visual-studio-2015/

Step 2: Host the Demo Service in IIS or Console application

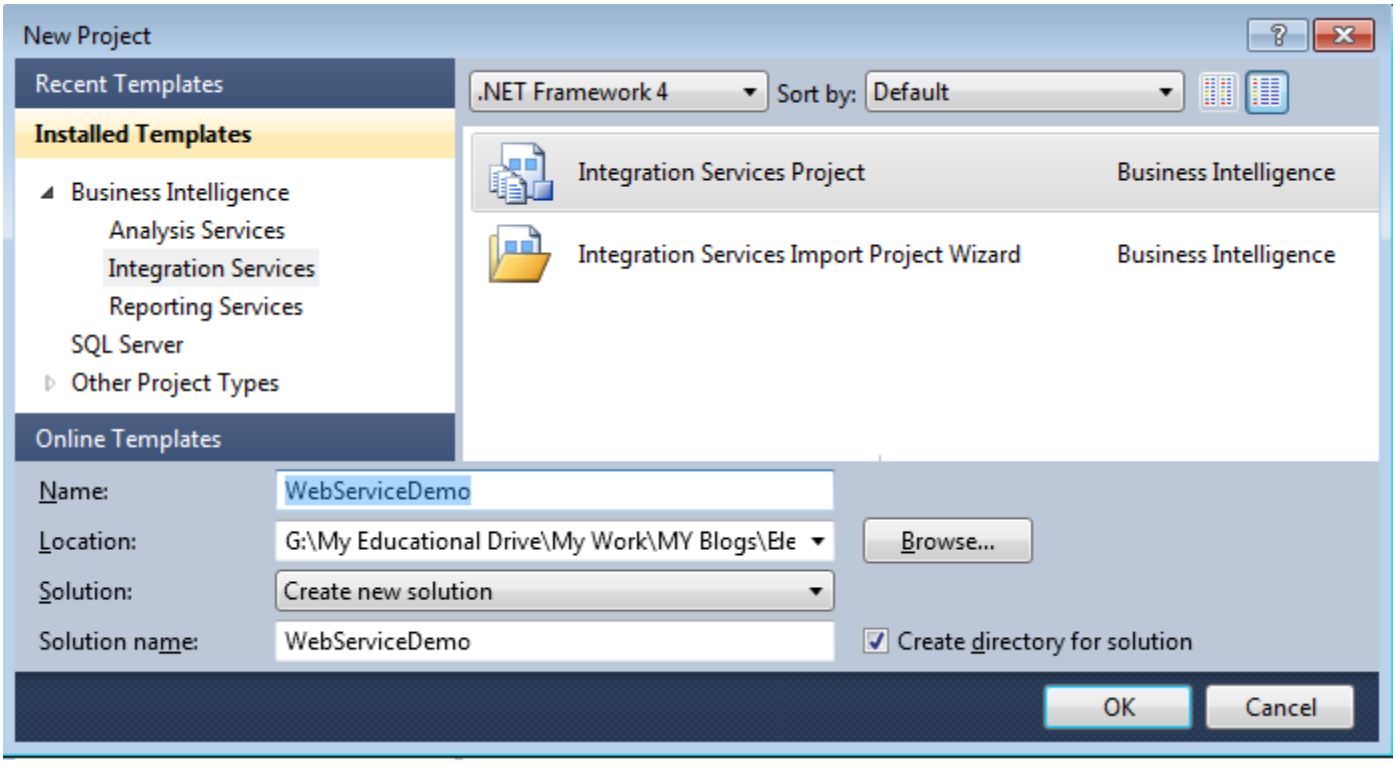
Please go through the following link for getting help of publishing a web service project.

Link: https://docs.microsoft.com/en-us/dotnet/framework/wcf/wcf-service-publishing

Note: we will use our published WCF project link: http://android.akij.net/AGWebService.asmx

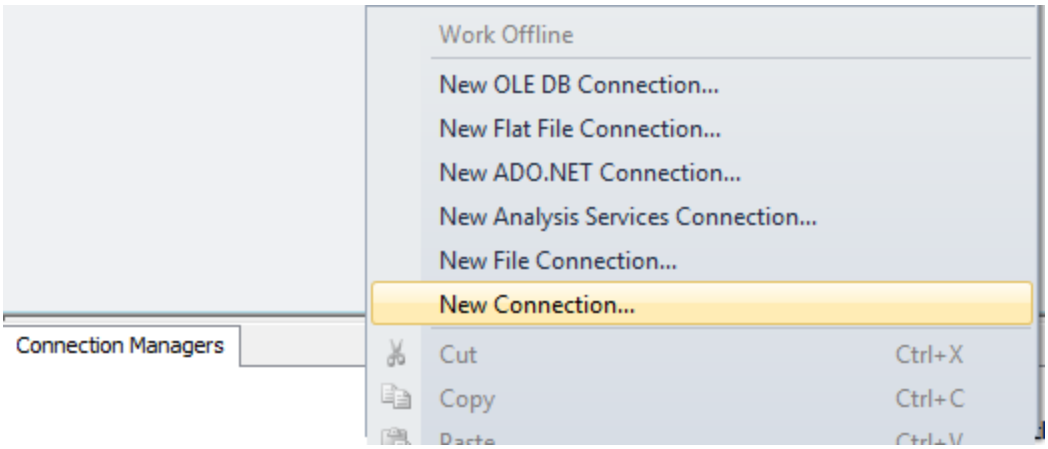
Step 3: Create new Integration services project

Open Sql Server data tools (or business intelligence development studio). Click File>> New>> Project. Select integration services from the left section of create project dialog box, Enter name of the project as WebServiceDemo and click OK

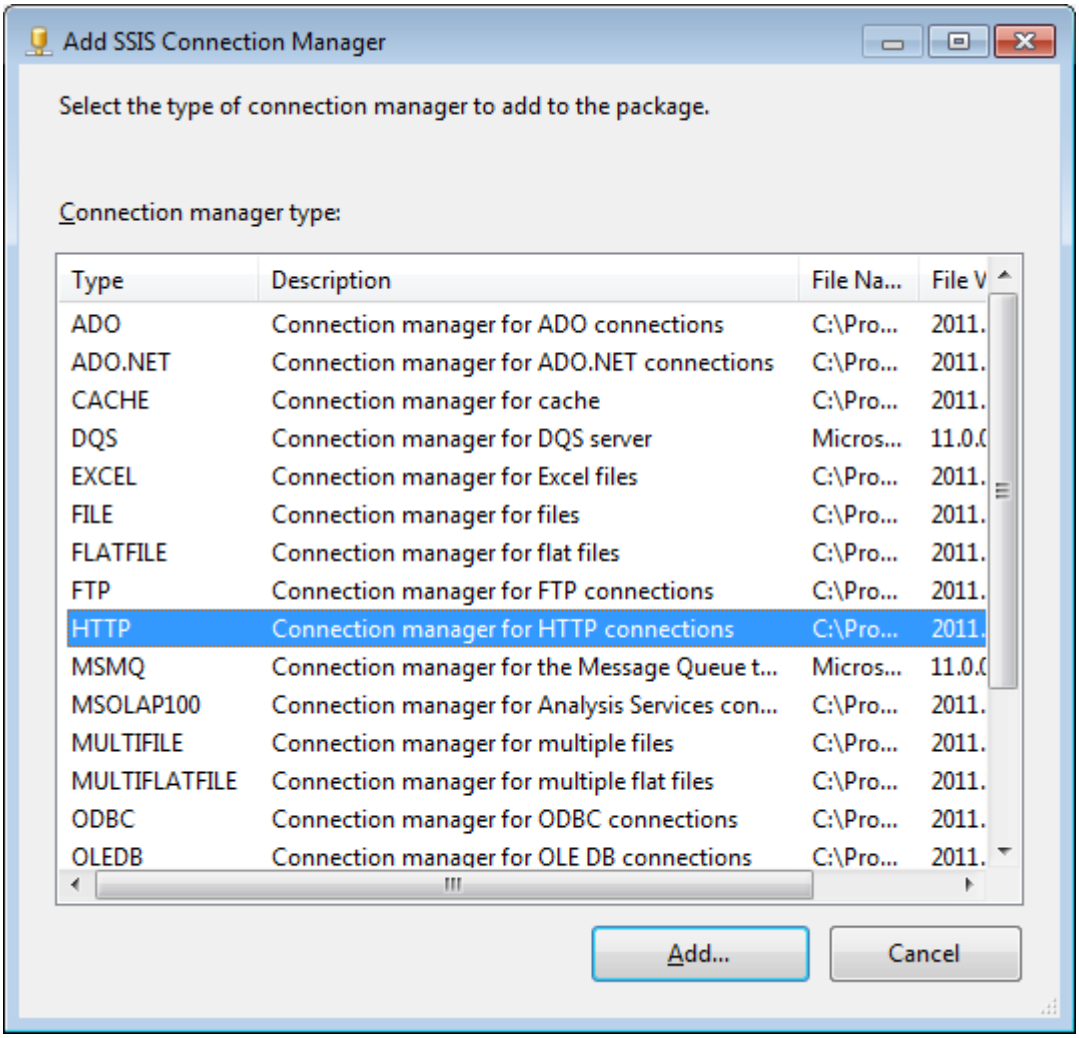


Step 4: Create HttpConnection Manager

1. Right click the connection manager in the bottom and select new connection.

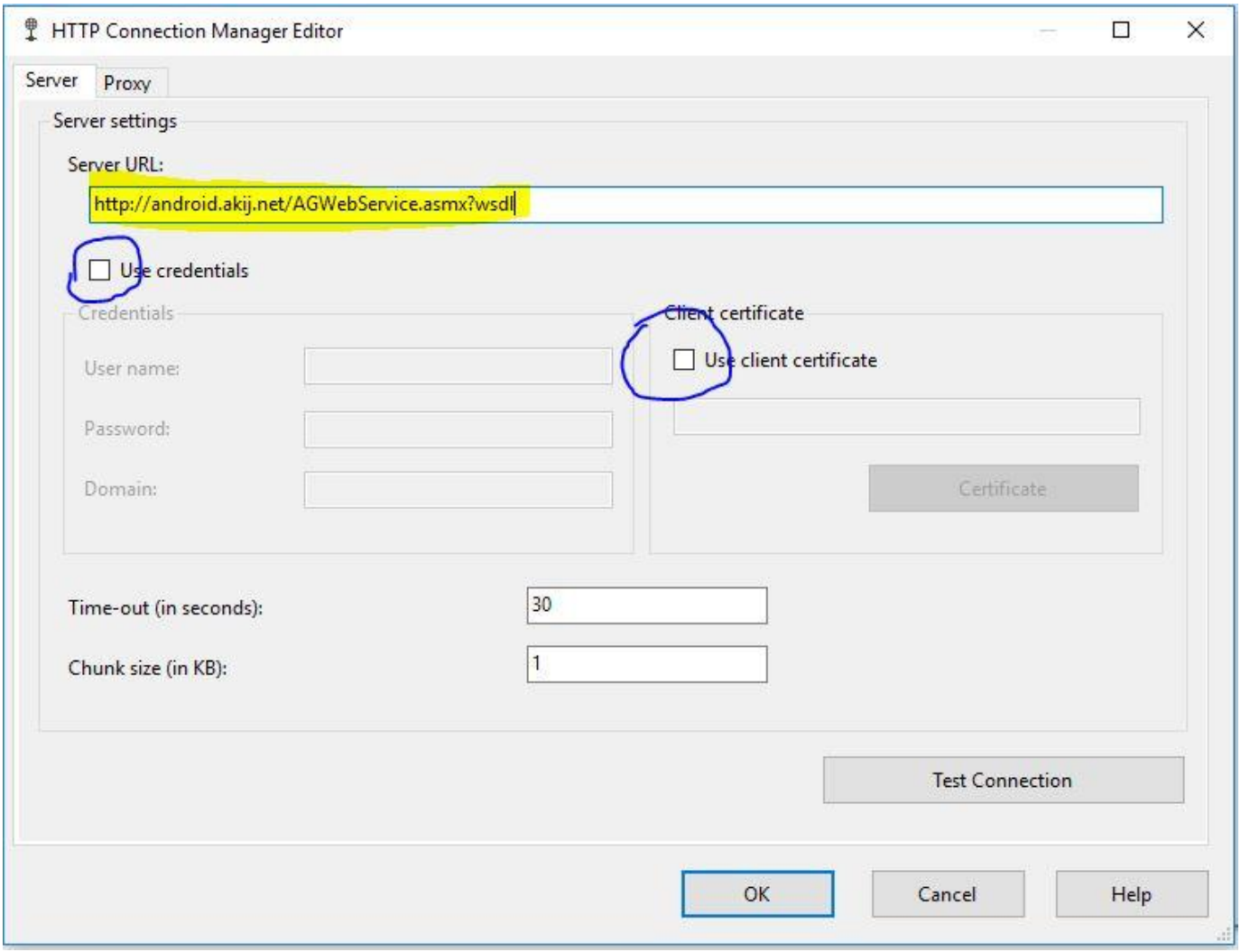


2. Select HTTP from the dialog box and click ok.



3. Put the address of your WSDL file in the next dialog box.

   Note: How to get the WSDL address is not your concern. The one who created the Service will give you that.

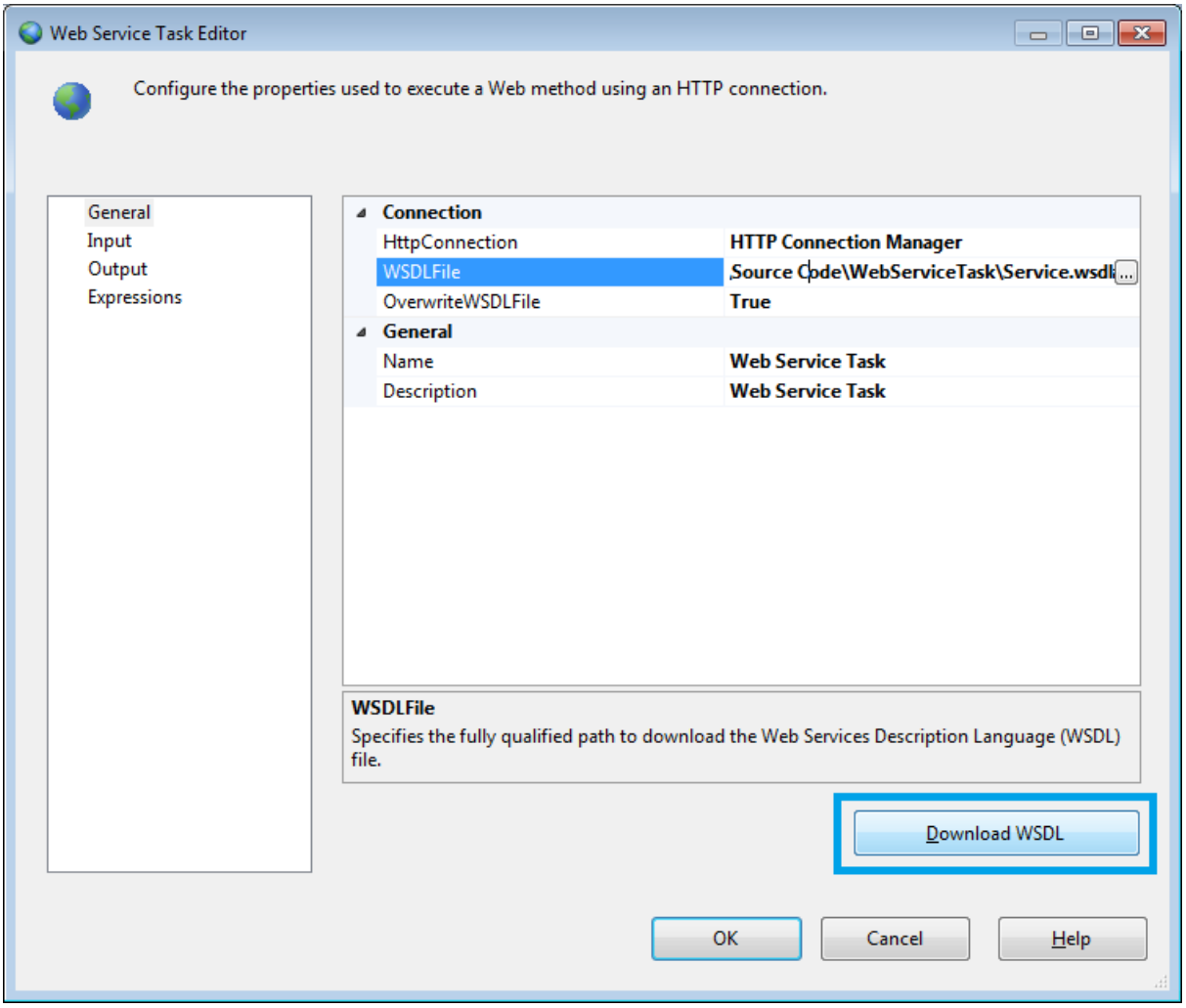


Note: Provide the Username and Password or certificate if service is accepting any. This information will also be available with you.

4. Click OK.

Step 5: Add Web Service task and configure it

1. Take WebService Task from the SSIS toolbox and add it to the SSIS designer tab.



Double click the task. It will open up the "Web Service Task Editor Dialog box".
Set the HttpConnection property to one set in the above step.
Set WSDL File property to some absolute file path.

Note: Don't select the path, rather type it because WSDL file is not available yet.

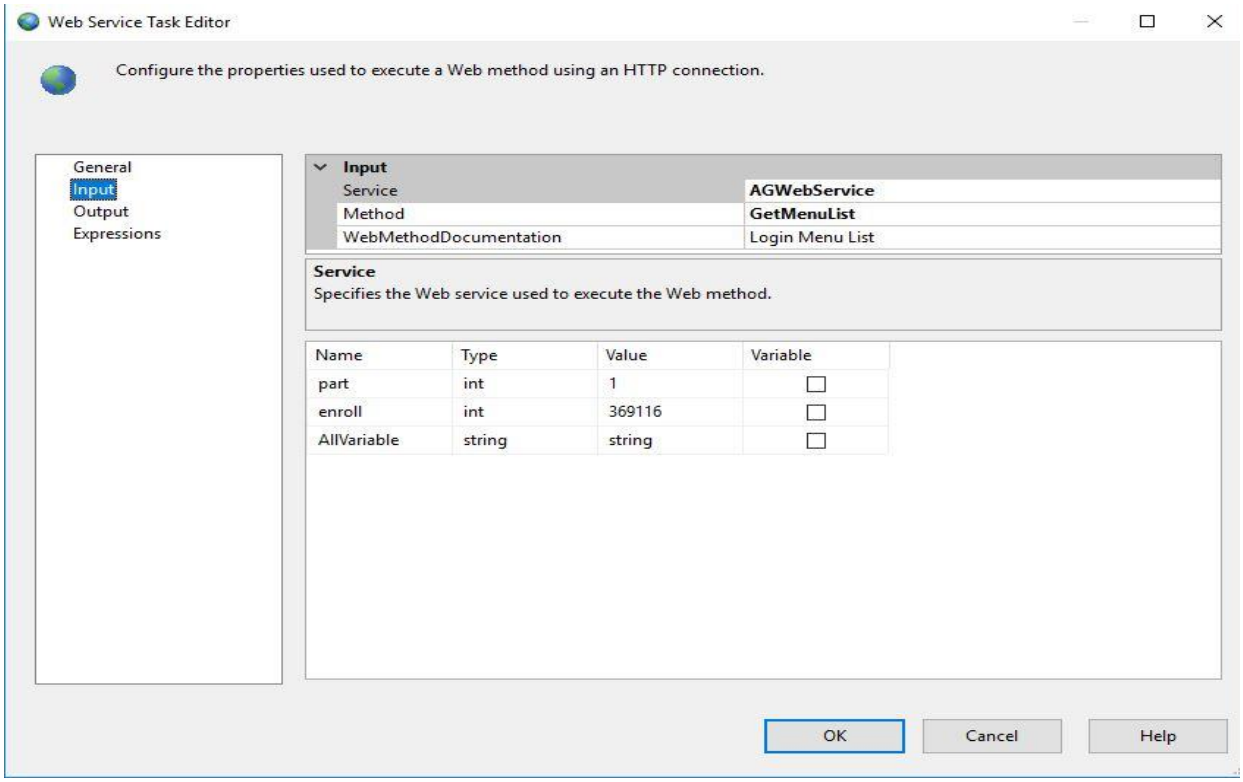
Path should be "Some_Valid_Folder_Path\AnyFileName.wsdl"

For example set it to "G:\BI Step by Step\3\Source Code\WebServiceTask\Service.wsdl"

In the bottom you will find a Download WSDL button. Click that. It will download the wsdl exposed by service into the path specified in above step.

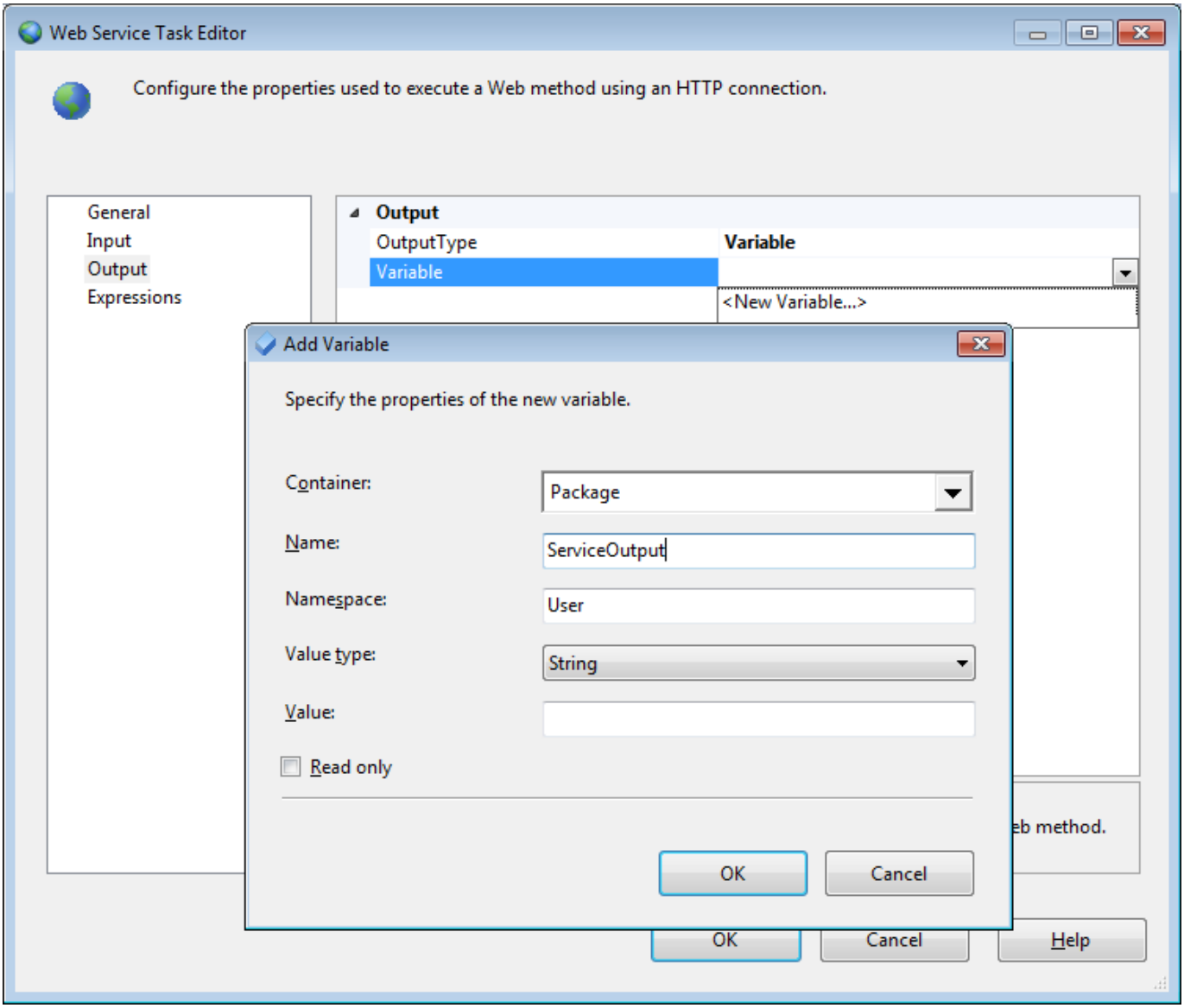
Now change selection from General to input in the left section.

Set service and method to one of the values in dropdown. Values had been added to both dropdowns when we clicked Download MetaData in the last step.



Step 6: Create Variable and Assign Value

Navigate to output section and set Output type to Variable. Next Set variable to “New Variable”. Enter Variable name as “ServiceOutput” and set type to String. Click OK to close the “Add variable” window. Click ok to close the “Web service task editor”.



Step 7. Add Script task to control flow

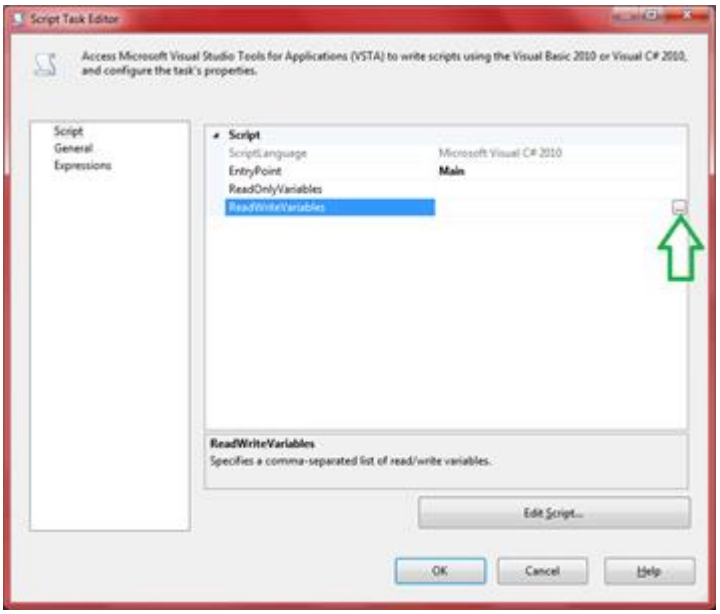
Take Script task from the toolbox and add it control flow

Step 8. Configure script task.

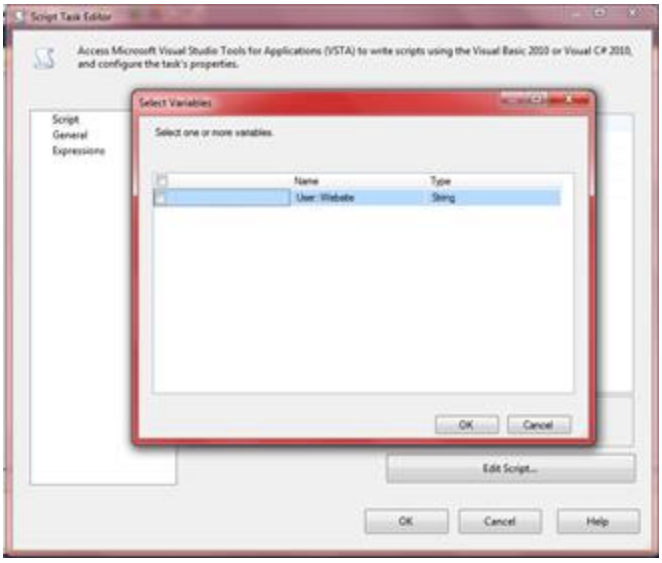
4.1 Double click the script task. It will open Script task editor.

4.2 Next we have to configure what variable we want to pass to the script. We can pass variable as either read only variable or read write variable. Click the triple dot button in front of either ReadOnlyVariable or ReadWriteVariable.

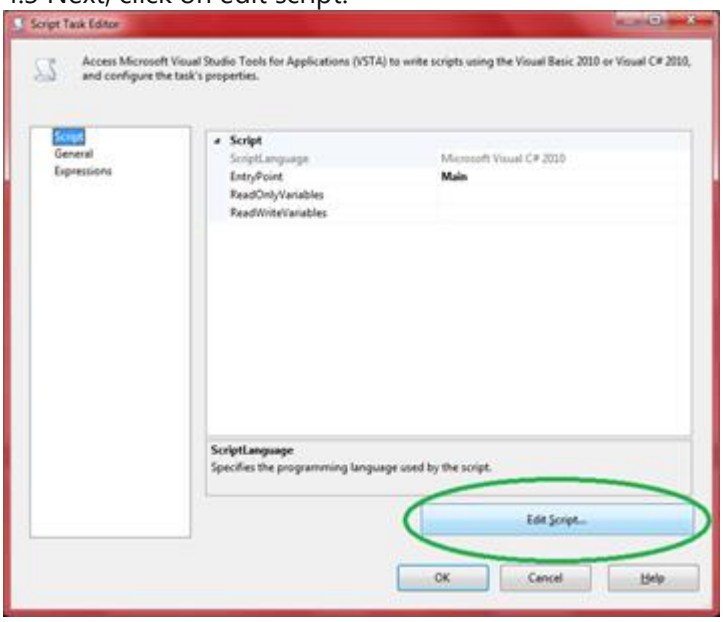
SSIS-Web Service With Xml and Error Handling in control flow



Select the variables and click OK.



4.3 Next, click on edit script.



It will make a new instance of Visual studio open up with a ".cs" extension file.

4.4 In the main Method write the following code.

```
public void Main()
{
    // TODO: Add your code here
    MessageBox.Show(Dts.Variables["Website"].Value.ToString());
}
```

4.5 Press Ctrl+Save and save the file.

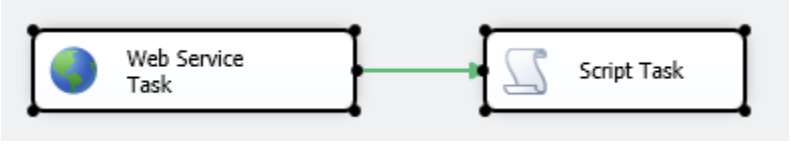
4.6 Close the Current Visual studio and go back to Sql server data tools where "script task editor" is open.

4.7 Click OK.

Add Script task from the SSIS toolbox and configure it to display the value of "ServiceOutput" variable in message box. Please refer day 2 to learn how to work with script task.

Step 9: Decide the execution flow.

Use the precedence constraint and make Webservice task executed first and then ScriptTask.

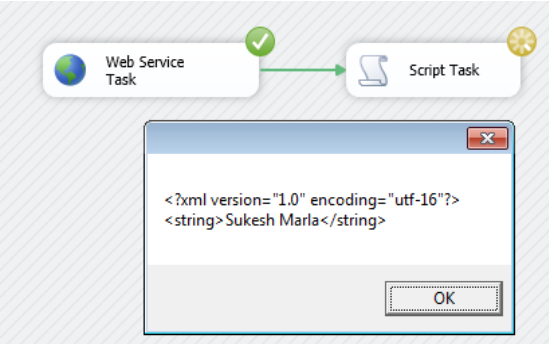


Click the Web service task. You will see a green arrow coming out of it. Take that line and connect it to ScriptTask.

Note: If we don't use precedence constraints here both the tasks will execute in parallel. With Precedent constraints we added a constraint "One will execute only after other executes successfully"

Step 10: Execute and Test

1. Make sure service is executing. (For demo you can use the WCF service attached in the article. From the WCF Service project and execute hosting project.)
2. Press F5 and confirm that package is working



Step 11: Add Xml Task and configure it

Take "Xml task" from the SSIS toolbox and add it to the control flow.

Step 12: Add variable for storing final result

Right click the designer section, select variables and add a variable called "XmlOutput"

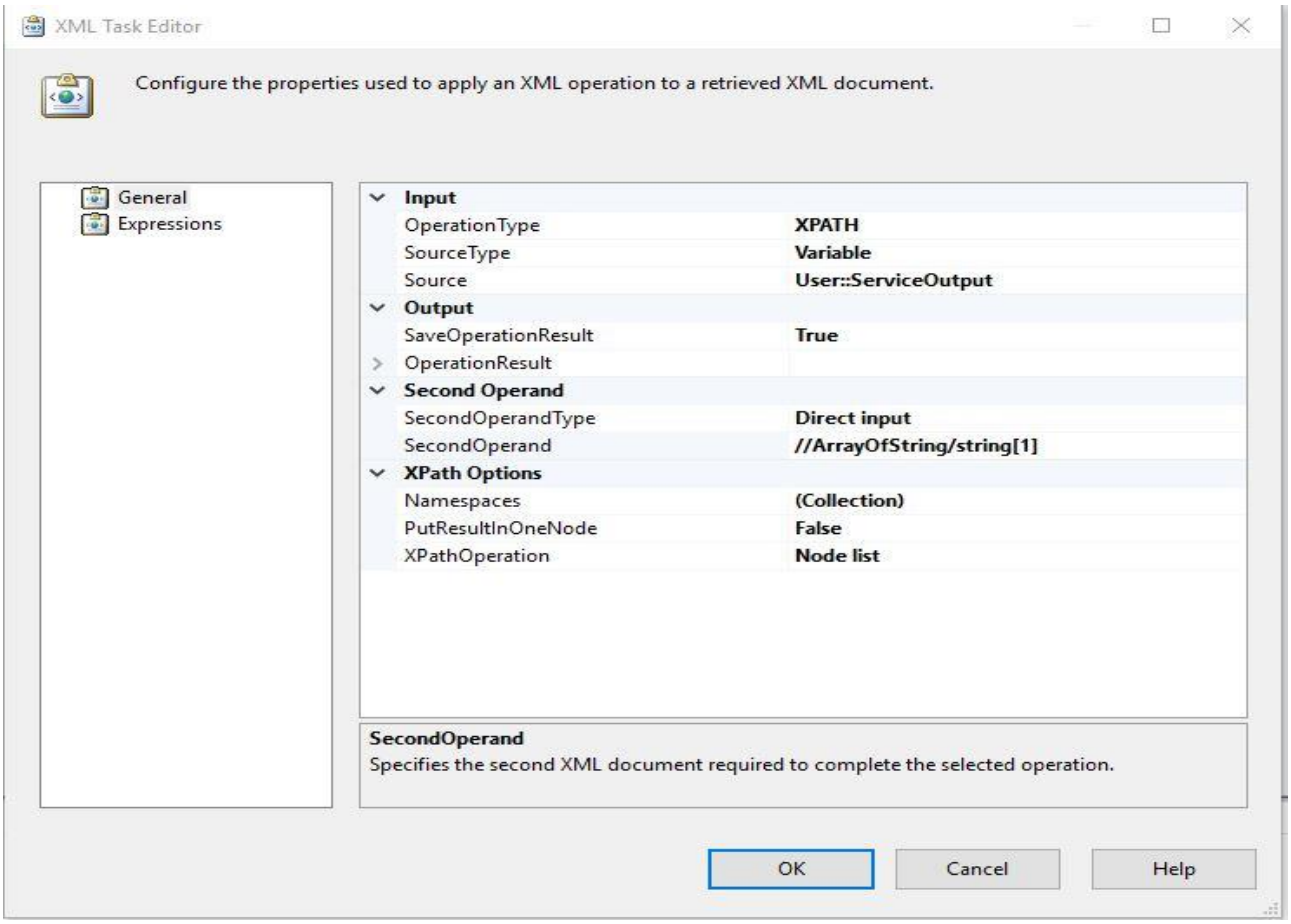
Step 13: Configure the XML Task

1. Double click the "xml task" it will open the "XML Task Editor window".
2. Set the properties as follows

I. I.Source Type to variable
II. II.Source to User::ServiceOuput (this variable will contain the webservice return value in xml format and it was created in last demo)
III. III.OperationType to XPATH
IV. IV.XPathOpearation to "Values"
V. V.SecondOperationType to DirectInput
VI. VI.SecondOperand to "//ArrayOfString/string[1]"

Note: If you want, you can store XPath expression in the file or variable and use it directly. For that set SecondOperationType as "File Connection" or as "Variable" accordingly.

VII. VII.SaveOperationResult to True
VIII. VIII.DestinationType to variable
IX. IX.Destination to XmlOutput (It was created in last step)
X. X.OverwriteDestination to True



3. Click OK

Step 14: Add and configure script task

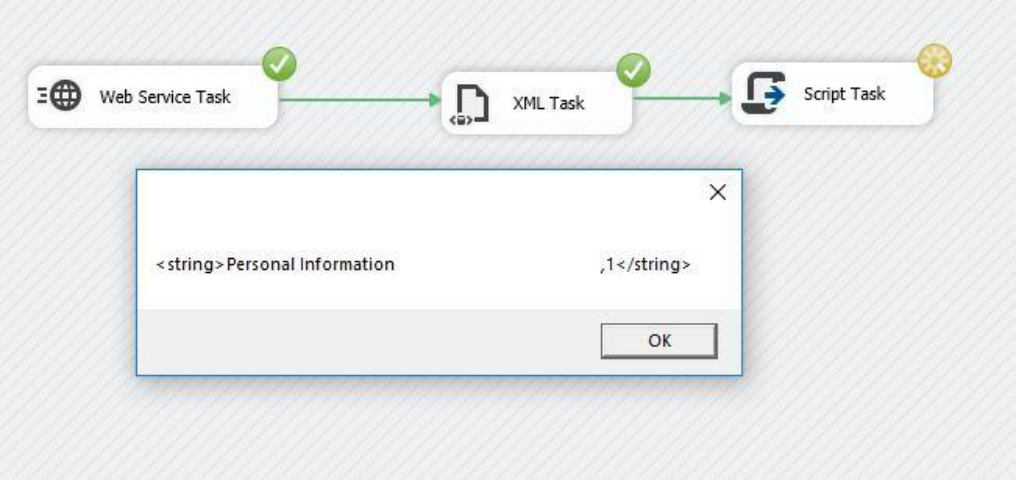
Add script task in the control flow and configure to display the value of XmlOutput variable in message box.

Step 15: Decide the execution flow

Use the precedence constraint and make “Web Service Task” executed first, then “Xml Task” and finally “Script Task”.

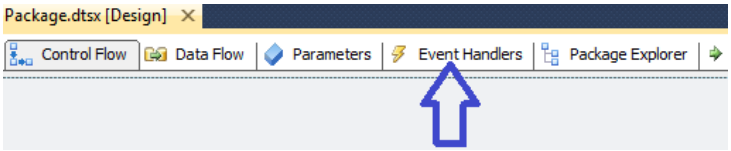
Step 16: Execute and Test the application

Press F5 and check the output



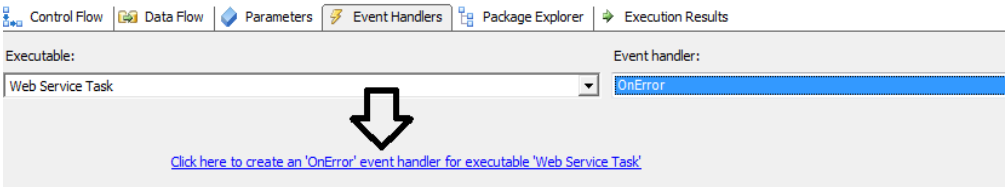
Step 17: Create Event Handler

I. Click the event handler tab in the SSIS designer window.



II.You will find two dropdowns in the top named “Executables” and “Event handler”. Select “Web Service Task” as Executable and select “OnError” as Event handler.

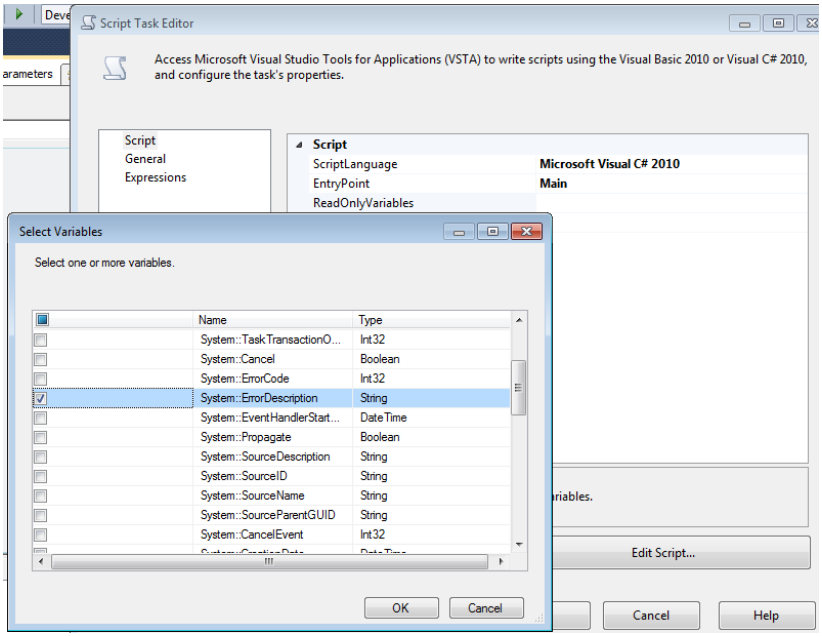
III.In the working area of designer window you will find a link “Click here to create an “OnError” event handler for executable “Web Service Task”. Click that



IV.That's it. It will create the event handler.

Step 18: Add Script task and configure it

Now in the working area of event handler add a script task.
Double click the task to open "Script Task Editor".
Make sure to select a system variable called "Error description" for ReadOnlyVariable property



Note:

"Error Description" is a system variable which will contain the latest error description in the SSIS Package.
Variable will be updated each time new error occurs.
We can use this variable as input for some other tasks.
Click the Edit Script and write a code to display the value of "Error Description" variable in message box.

Hide Copy Code

```
MessageBox.Show(Dts.Variables["ErrorDescription"].Value.ToString());
```

Step 19: Execute and test the application

Press F5 and confirm that everything is proper.