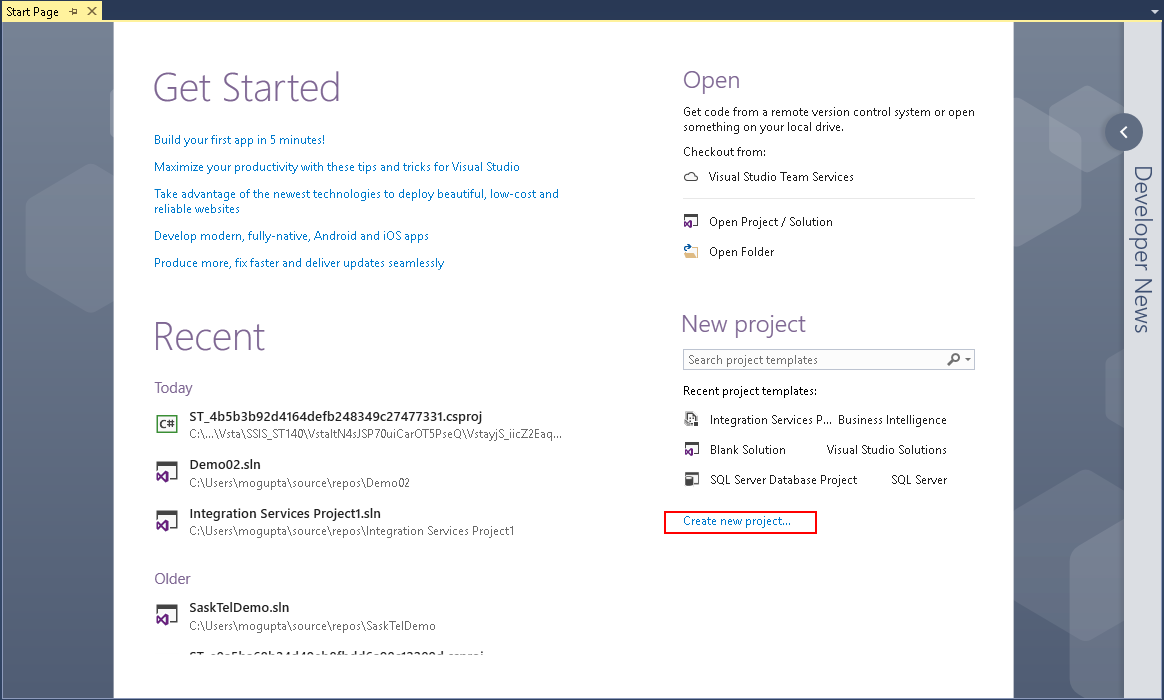
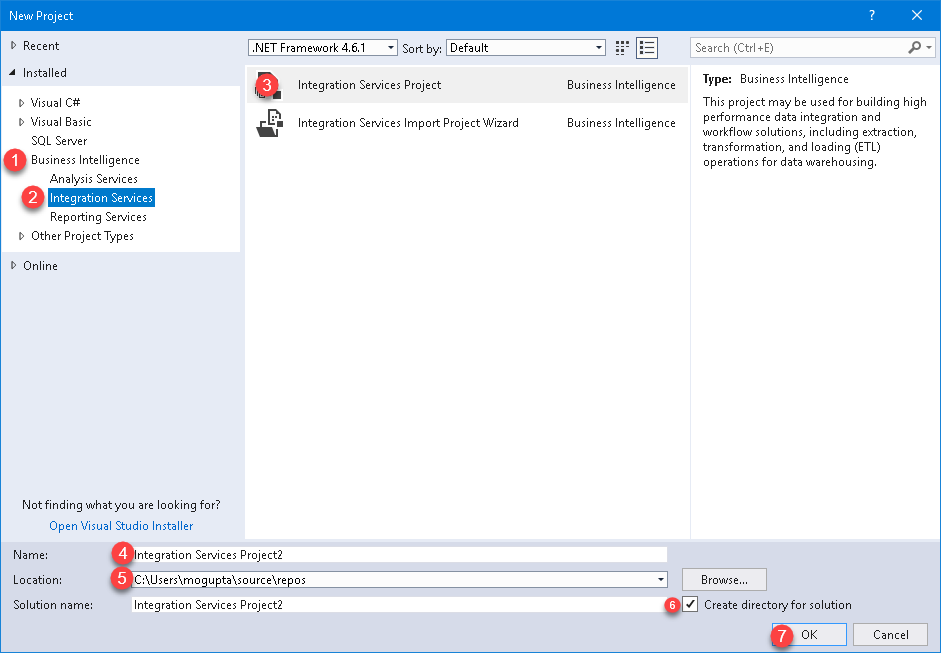
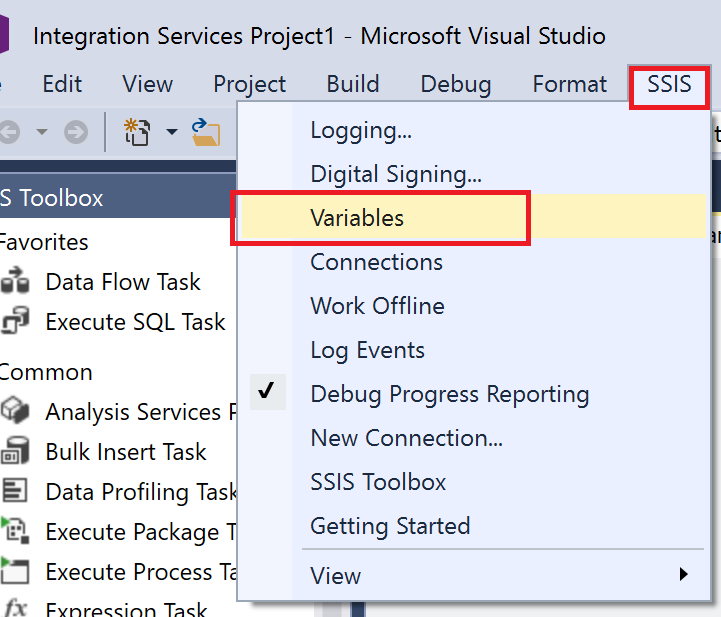
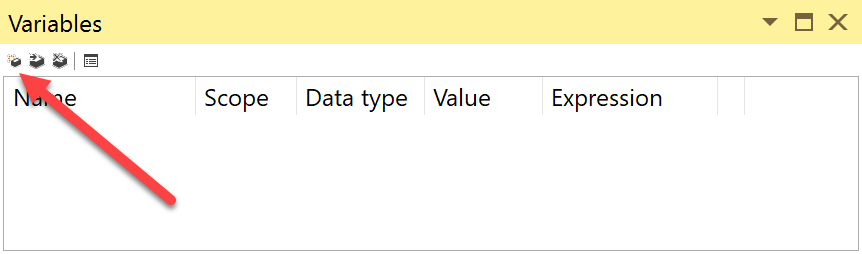
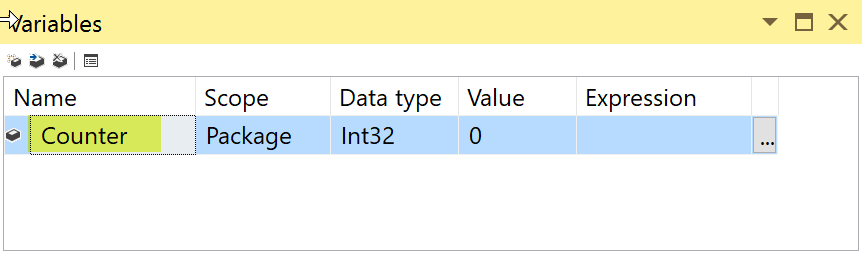
SQL Server Integration Services

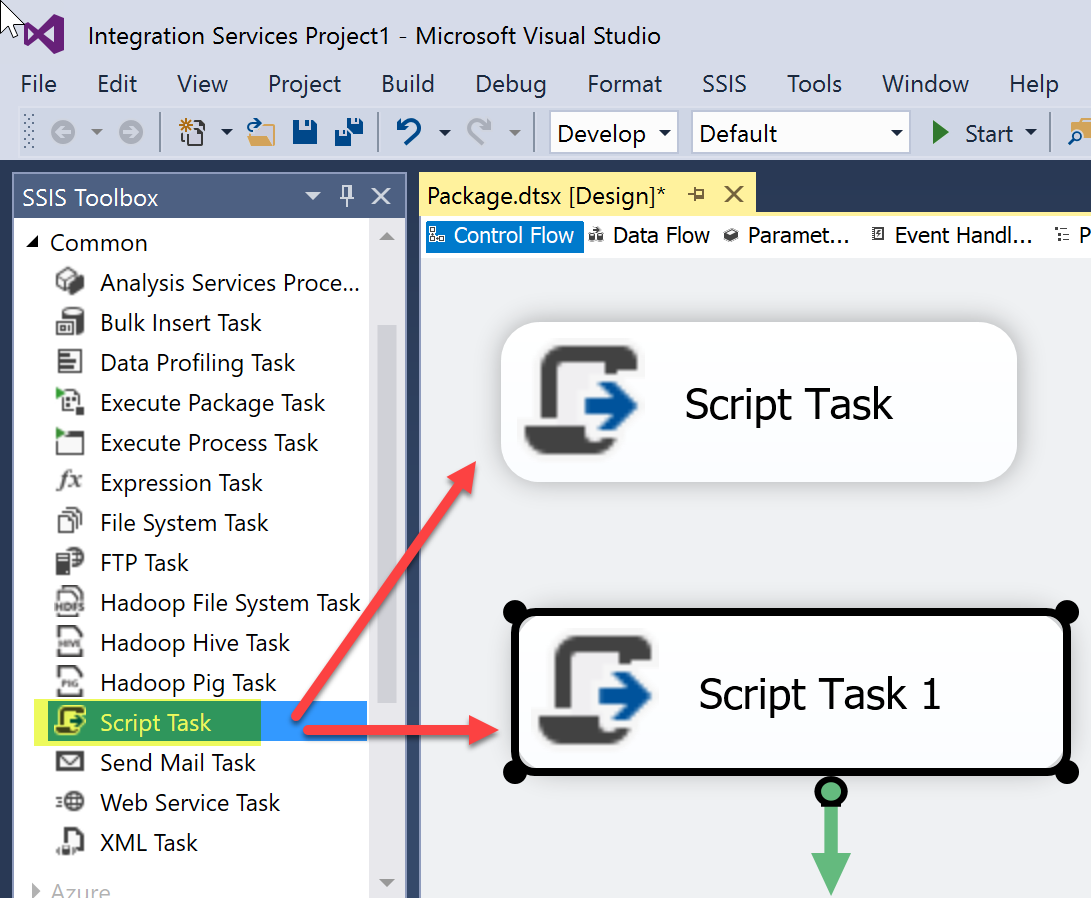
# Module 03: CONTROL FLOW: **SCRIPT TASK**

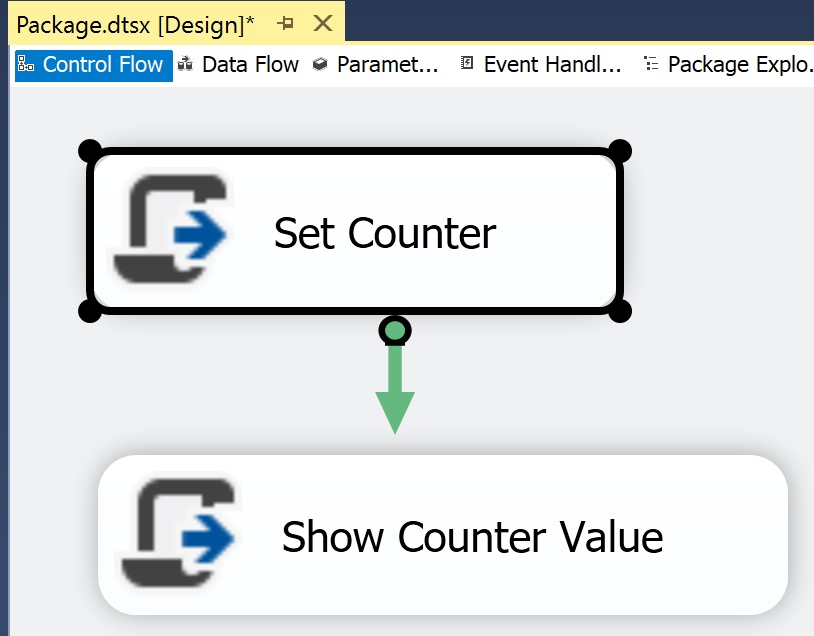
1. Launch SQL Server Data Tools (SSDT), under start menu look for Visual Studio 2017 (SSDT).
2. In the Start Page, click Create new project.  
   
3. In New Project dialog box select Business Intelligence > Integration Services > Integration Services Project. On the bottom enter in project name and location you wish to save the project. Make sure “Create directory for solution” is selected and click OK.



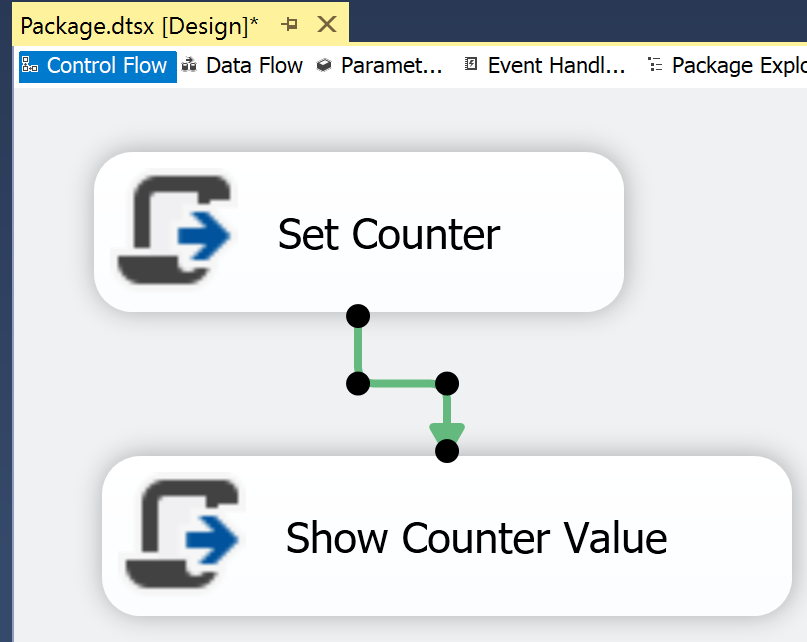
1. In the SSIS Menu select Variables   
   
2. Click on the New Variable icon  
   
3. Rename the Variable to Counter. Notice initial value is defaulted to zero.



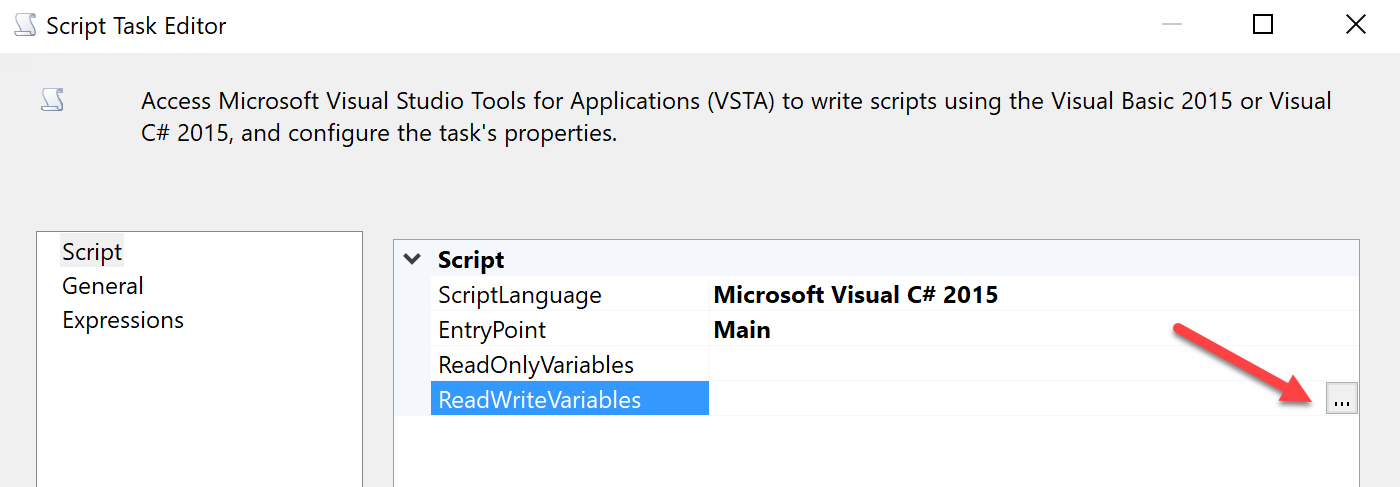
1. Drag and drop the Two Script Task onto the Control Flow canvas.  
   
2. Rename Top script task to “Set Counter” and Bottom task to “Show Counter Value”



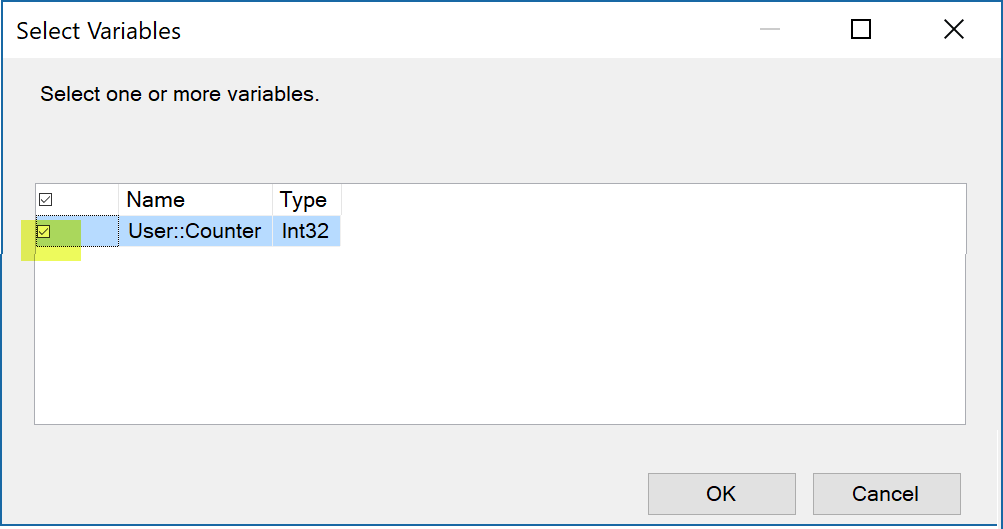
1. Drag the Green Precedence Constraint Arrow from the first task to the second task.



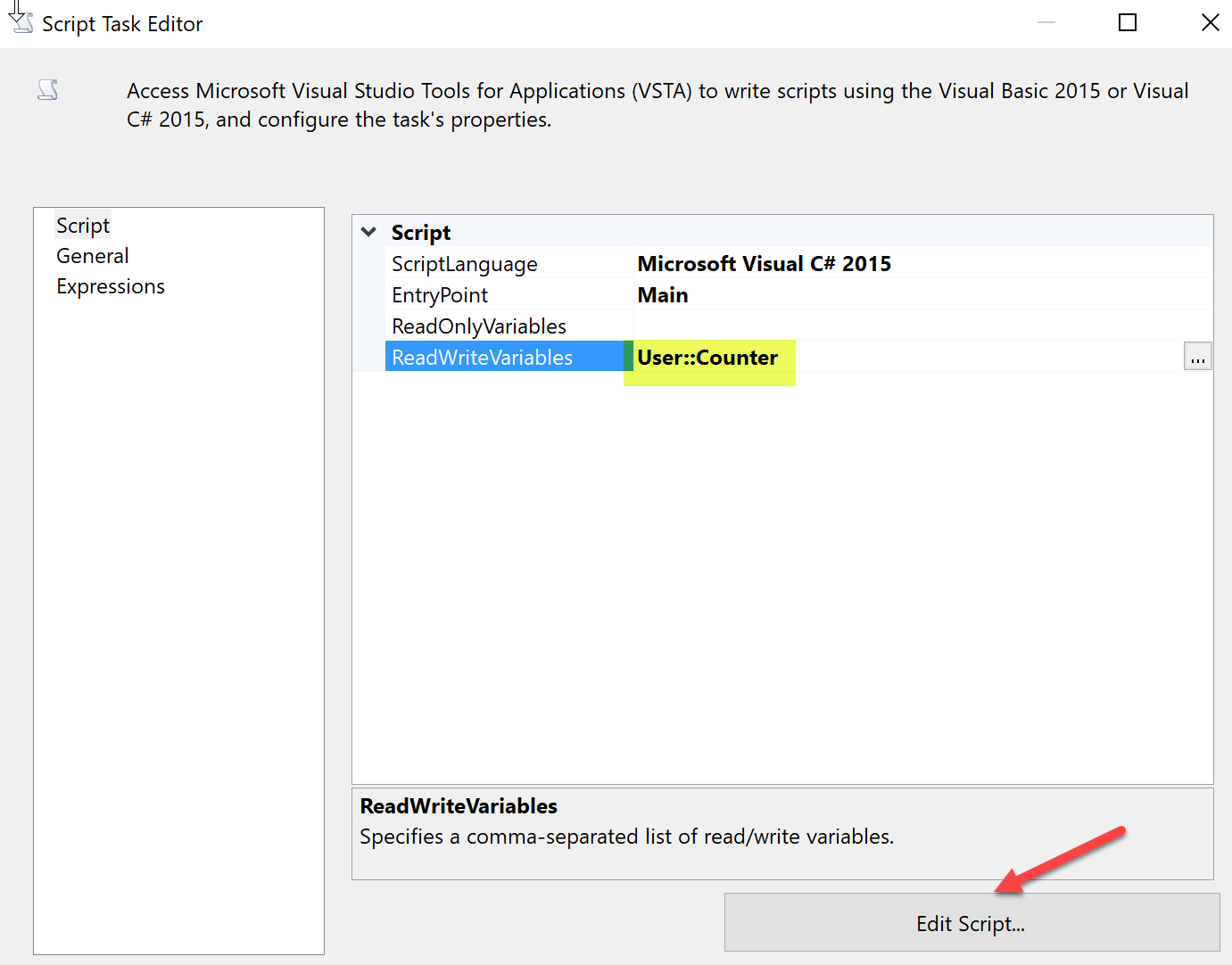
1. Double-Click “Set Counter” Task, since we will be assigning a value to the variable within the script task click on the ellipse button for the ReadWriteVariables



1. Select the Variable you want to modify, in this case Counter, then click OK.

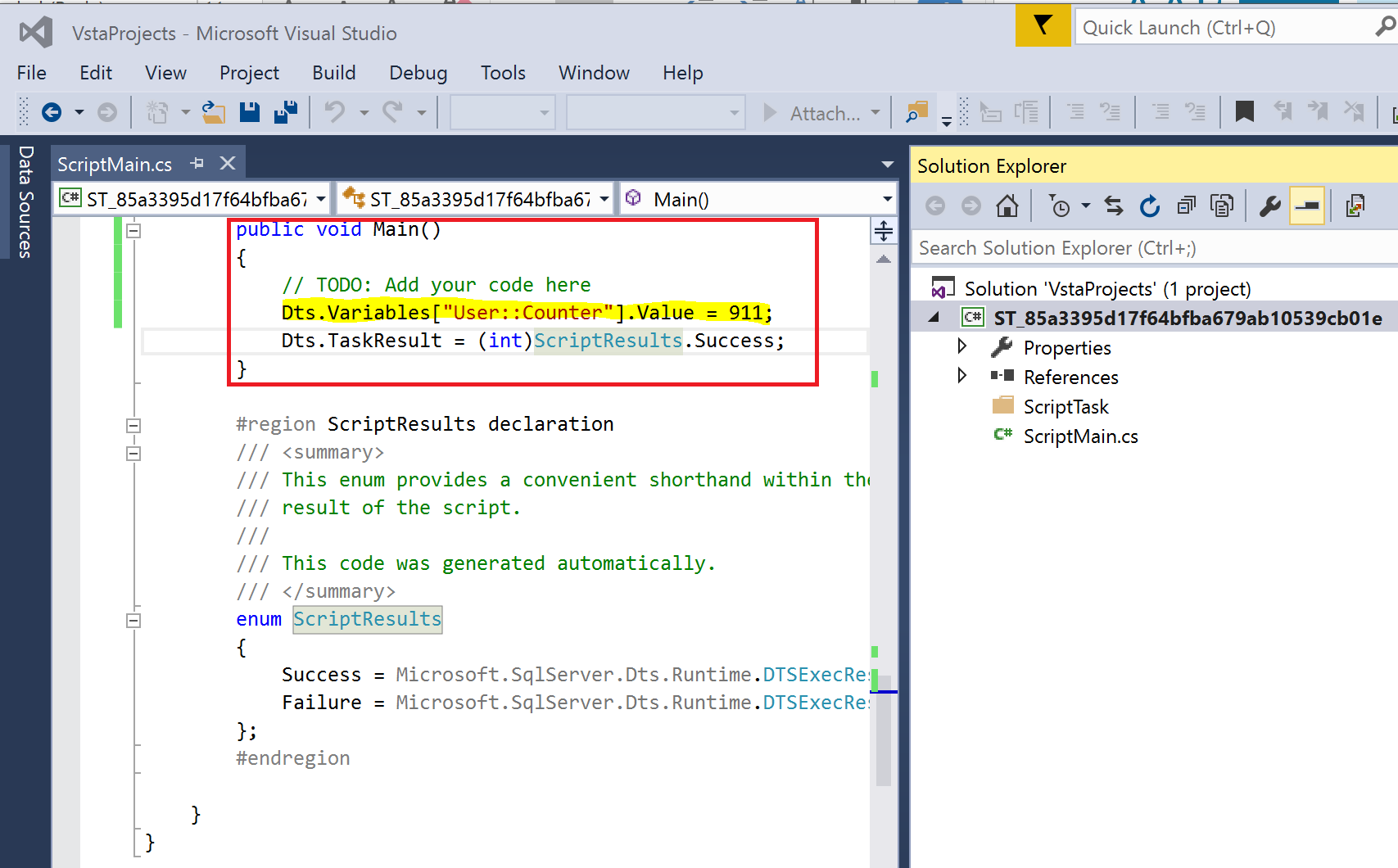


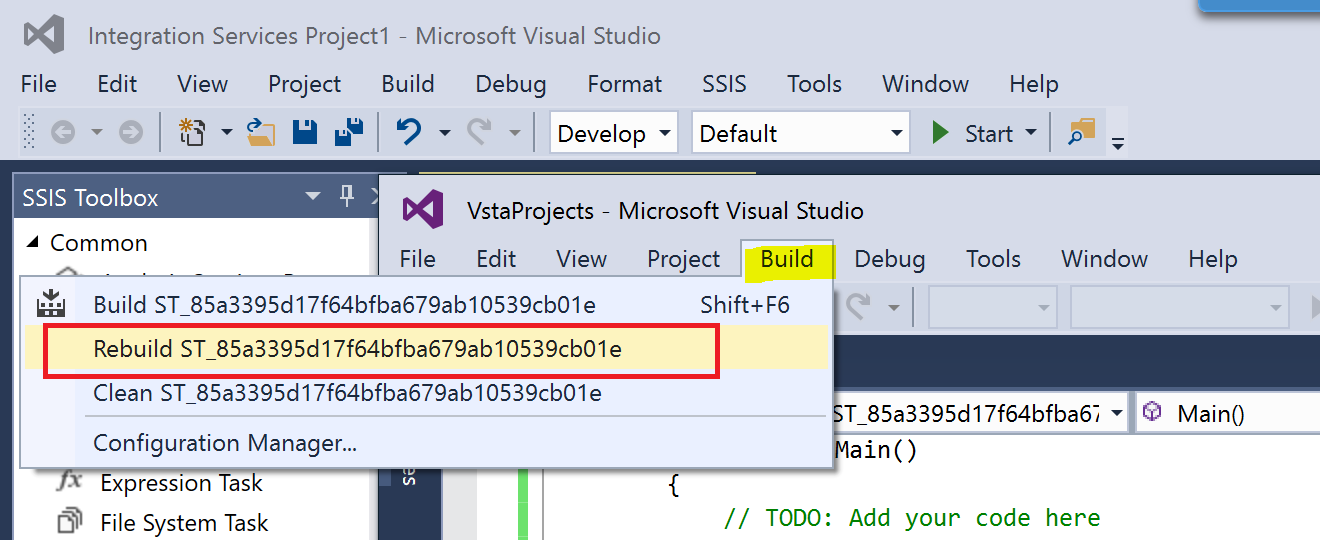
1. Click on the Edit Script Button

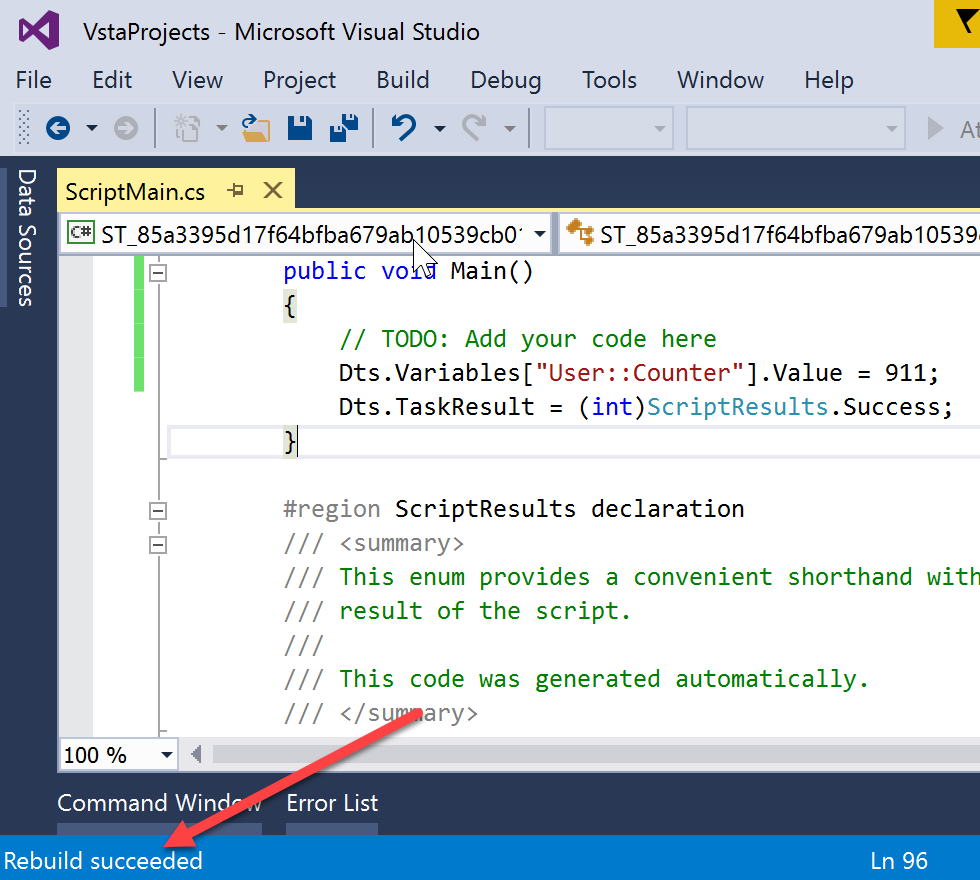


1. A Visual Studio Tools for Applications (VSTA) Scripting Windows will open, scroll down toward the bottom and the following code line to assign a value to in the Main() code block.

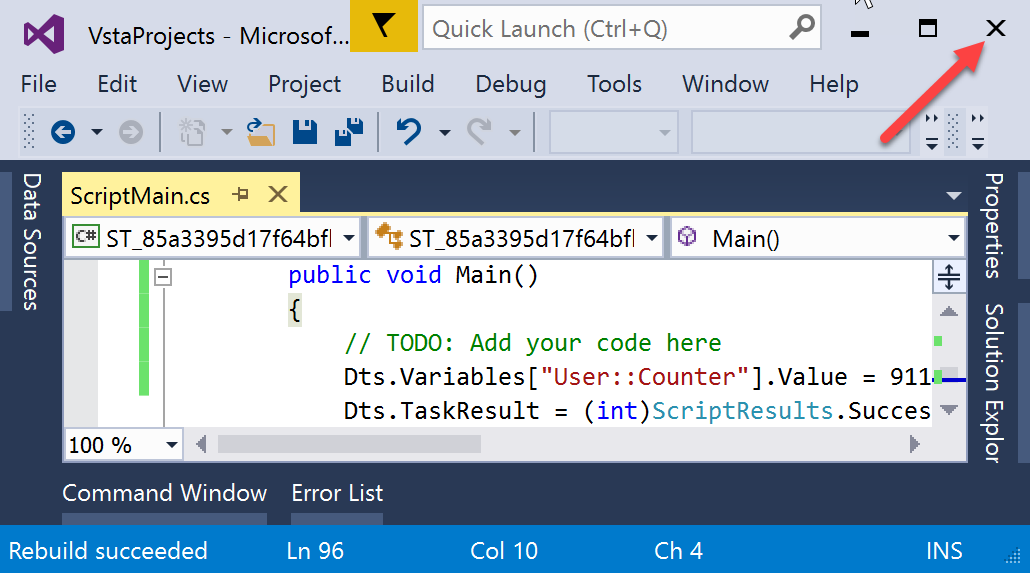
Dts.Variables["User::Counter"].Value = 911;



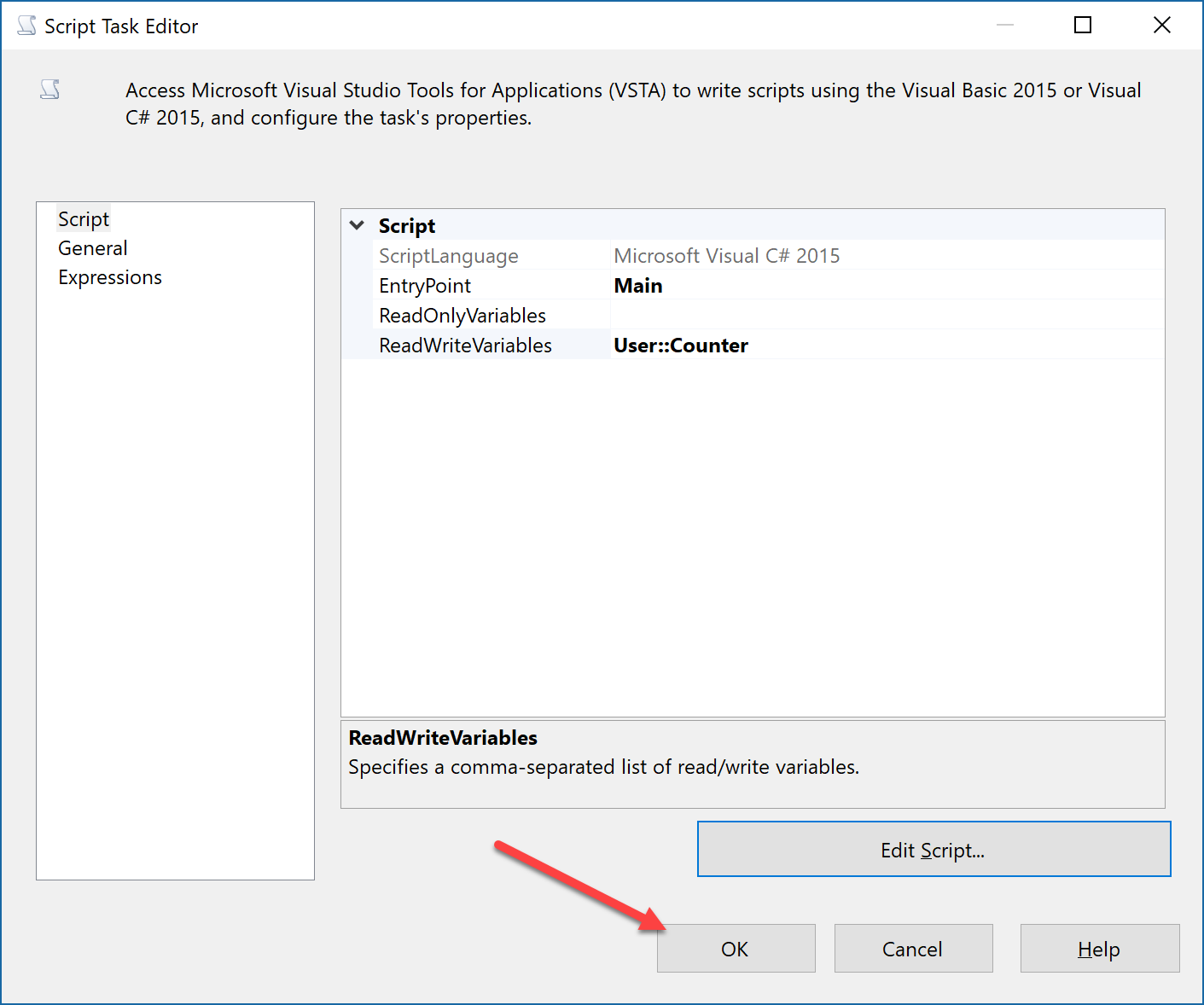
1. Ensure that you code syntax is correct, by verifying you can rebuild successfully. Click on Build menu and then select Rebuild.



1. Exit the the VSTAProject window

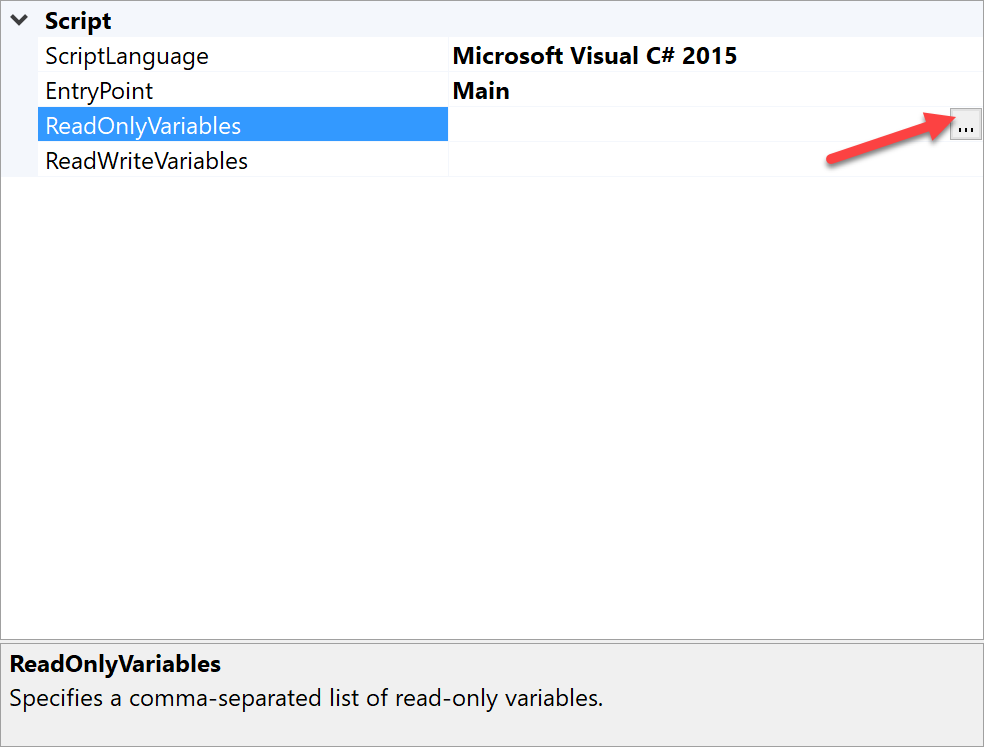


1. Click OK.

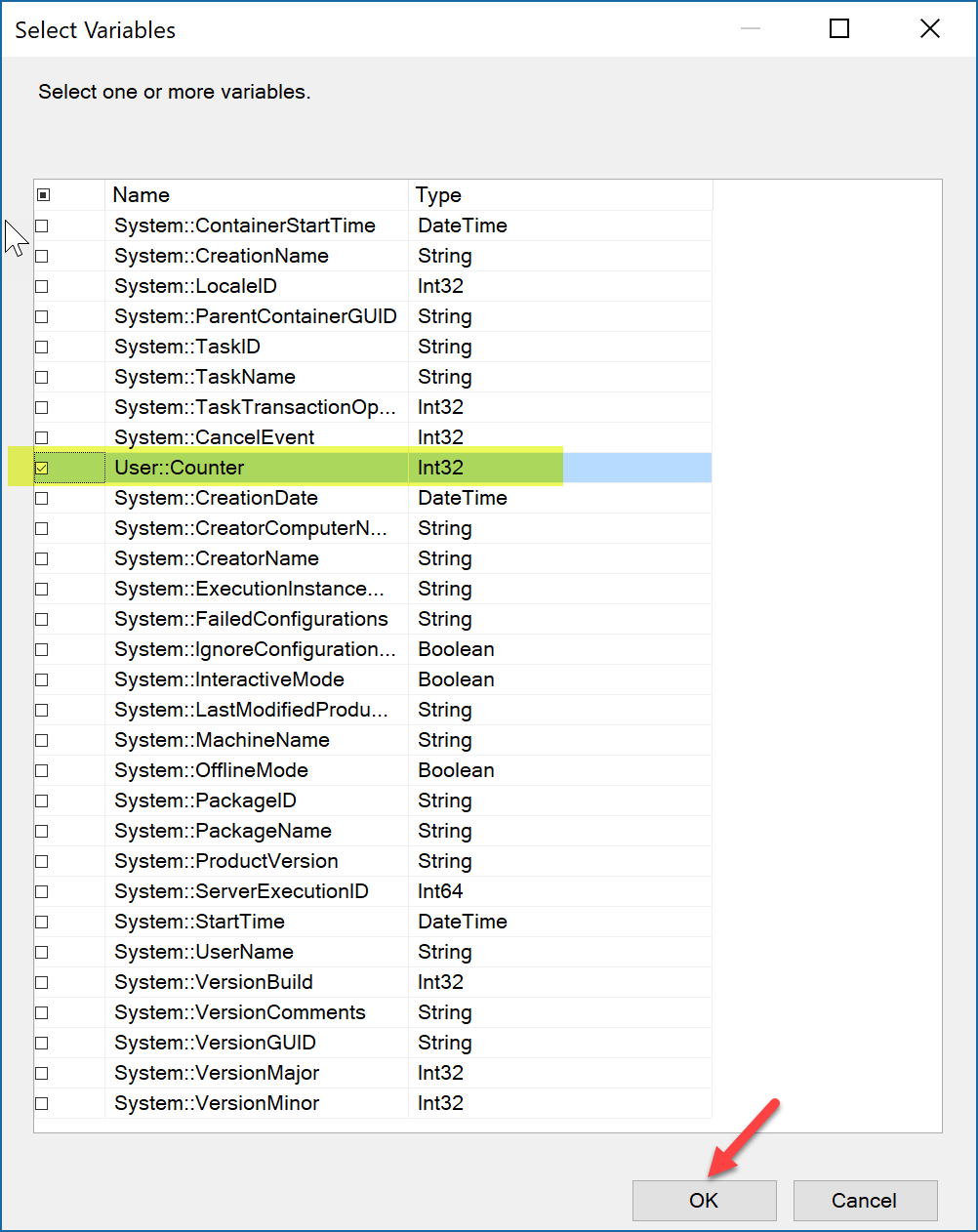


We will now set the next script task properties.

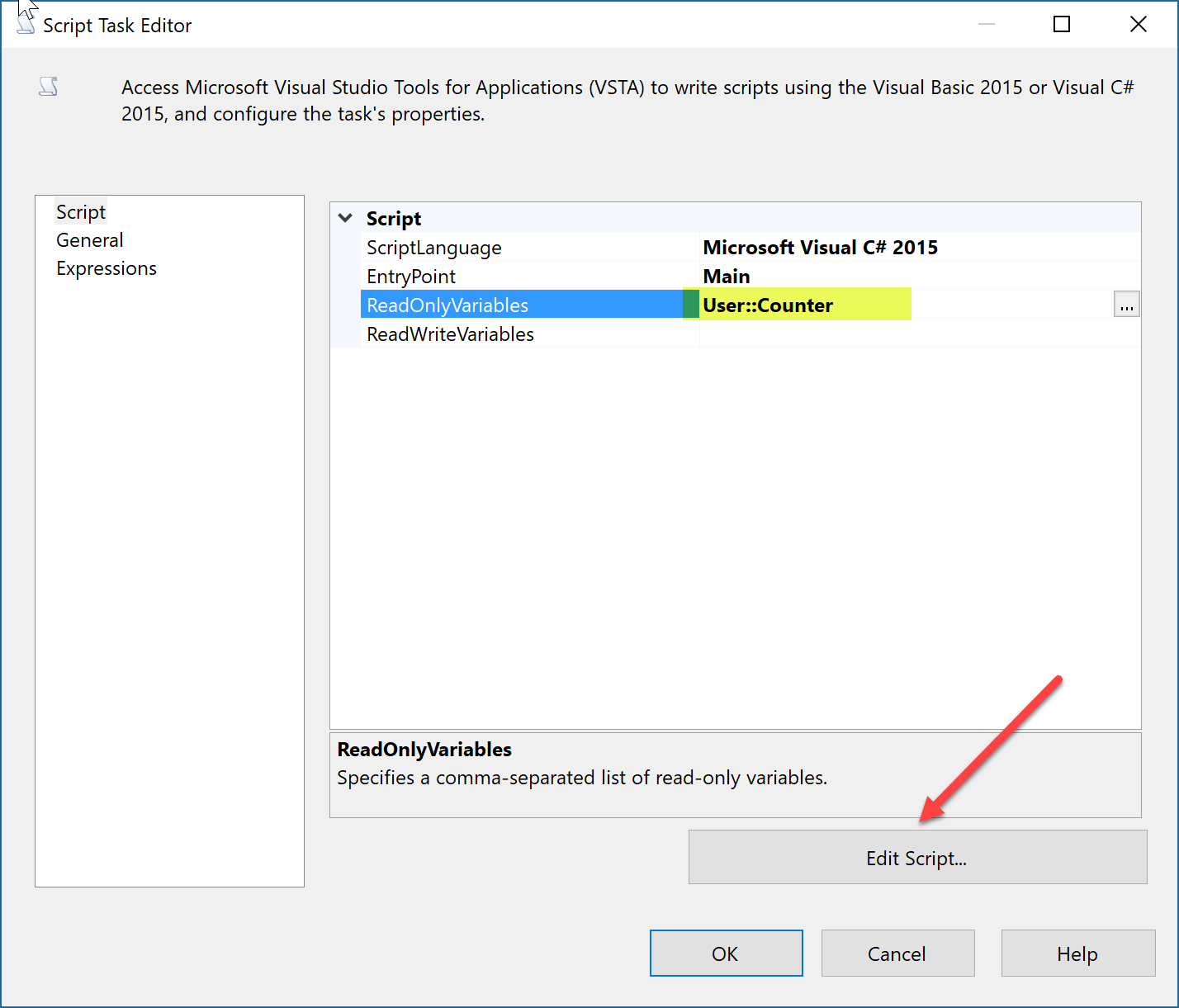
1. Double-Click the “**Show Counter Value**” Script, and then click on the elipse button for ReadOnlyVariables.



1. Select the variable you want to retrieve values for, in our case Counter, then click OK.



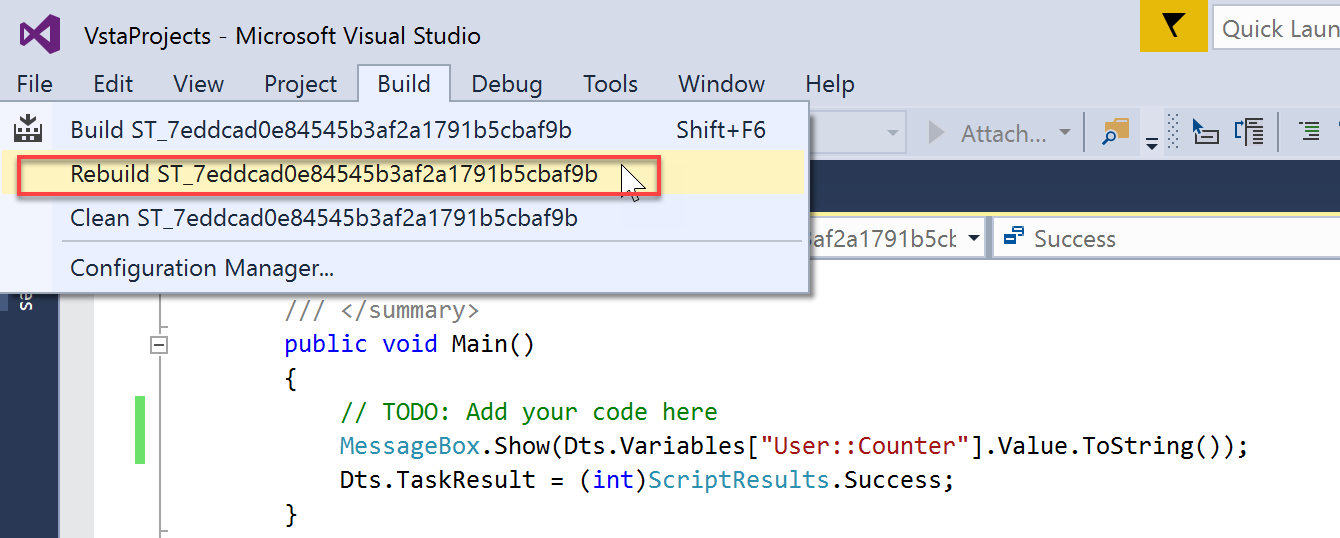
1. Click on the Edit Script button.

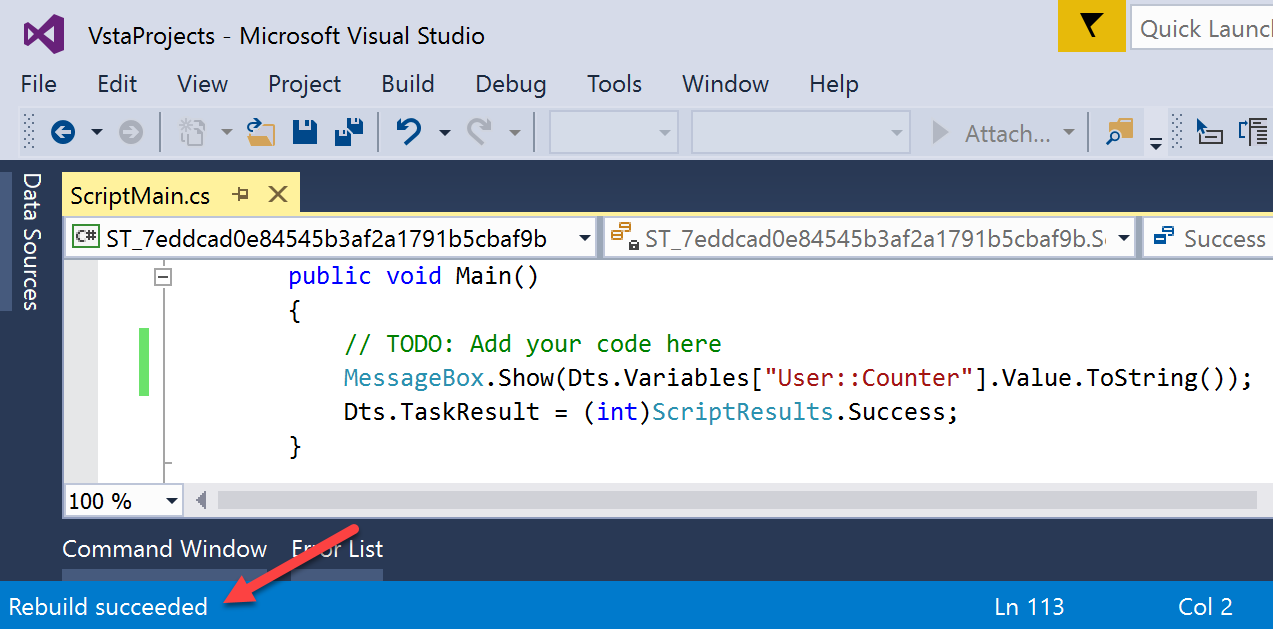


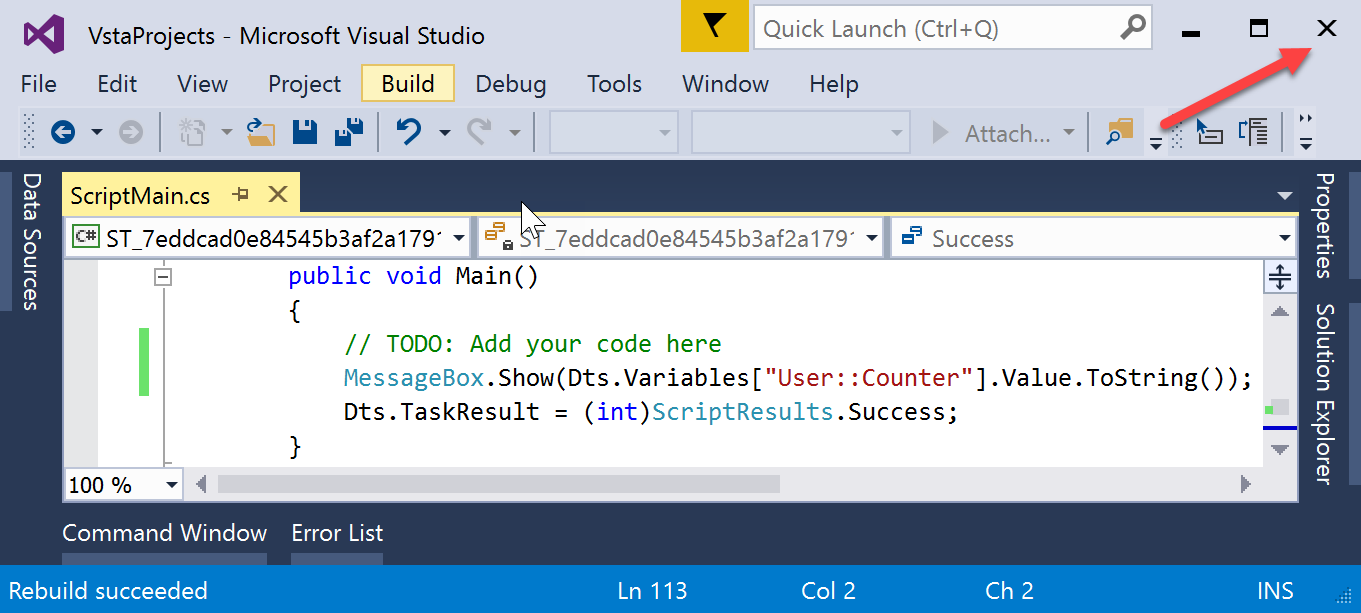
1. In VSTA window, Add the following line in the Main() code block. This will display the value of Counter in a prompt window.

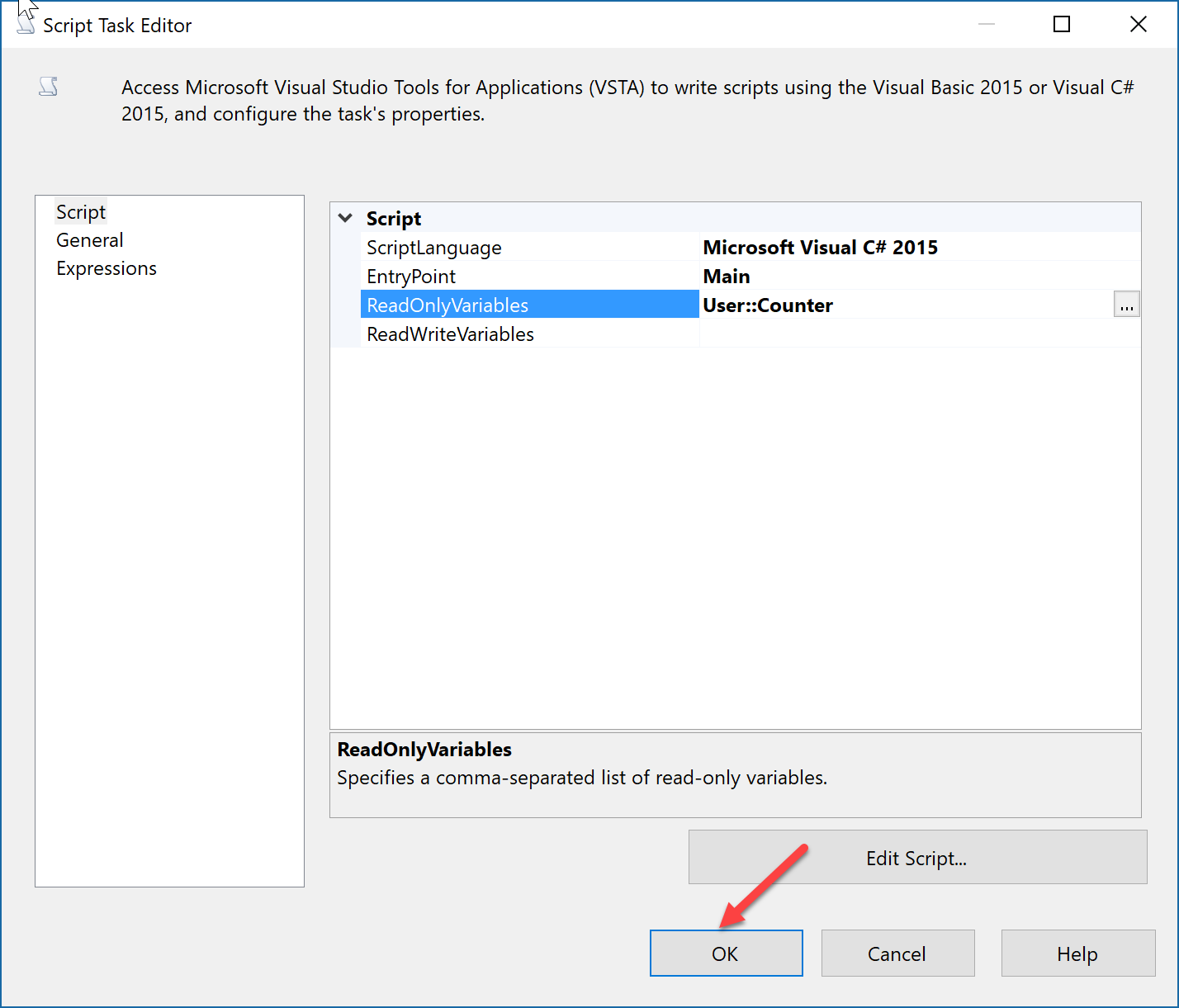
MessageBox.Show(Dts.Variables["User::Counter"].Value.ToString());

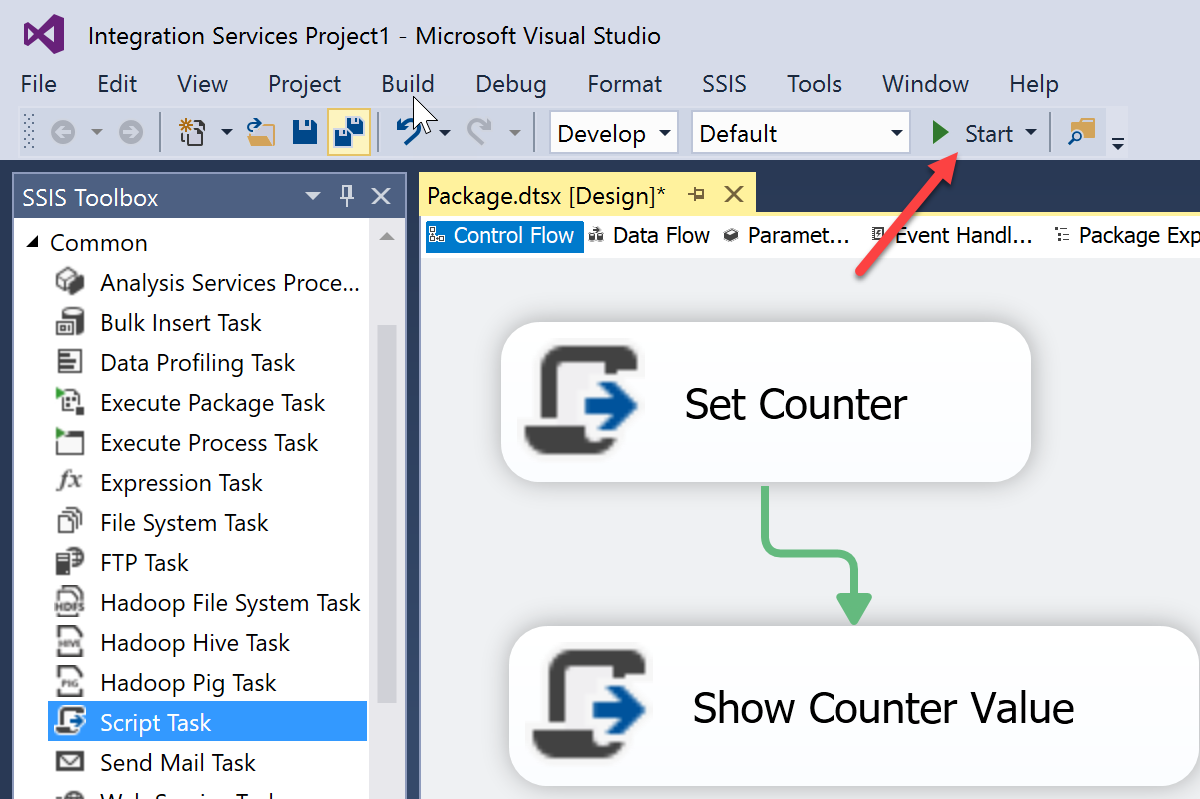
1. Rebuild to ensure there are no compile errors.





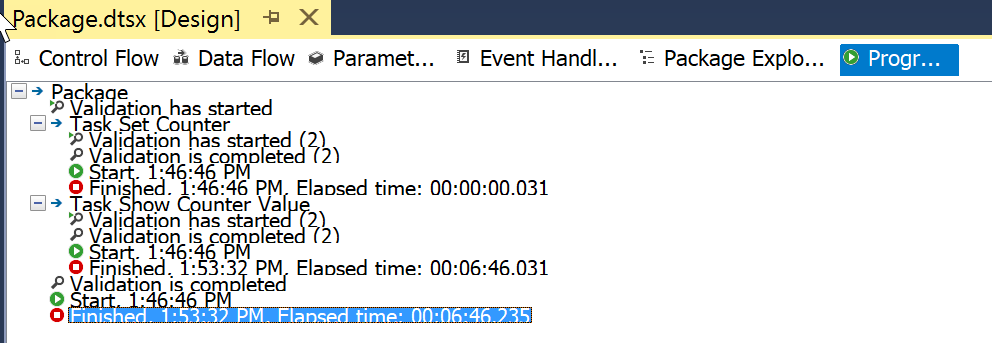
1. Close the VSTA window.   
   
2. Click OK.



1. Click Start Button.
2. Notice Package is Running and displaying the assigned value. But is waiting for user interaction (clicking on OK) for the package to complete. Click on OK.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Tasks | Counter Value | | Initial assigned when variable declared | 0 | | Set to 911 within “Set Counter” script | 911 | | “Show Counter Value” script  Correctly show the new value set | 911 | |  |

Look at the Progress Tab, and notice the execution time is approx. 6 minutes 46 seconds.



Let’s find another alternative to not having to wait.

1. Modify the “**Show Counter Value**” Script, and

Replace following code

MessageBox.Show(Dts.Variables["User::Counter"].Value.ToString());

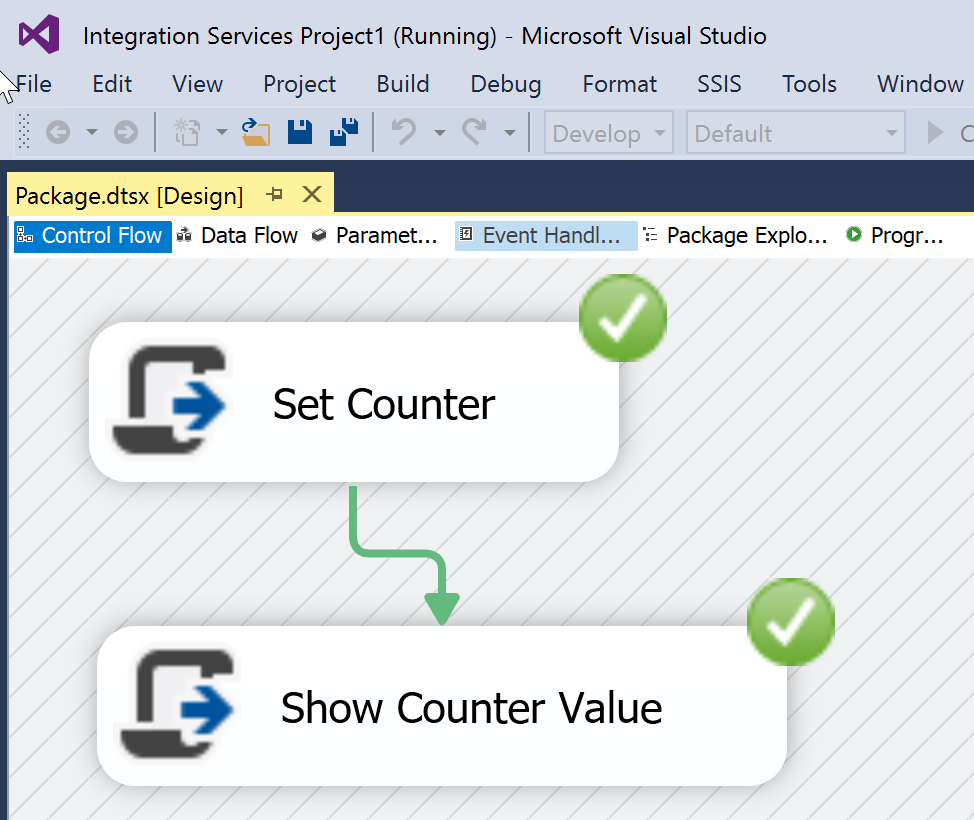
With the following code

bool fireAgain = false;

string description = "Counter Value= " + Dts.Variables["User::Counter"].Value.ToString();

Dts.Events.FireInformation(0, null, description, null, 0, ref fireAgain);

1. Then run the Package again.



1. Look at the Progress tab now.

