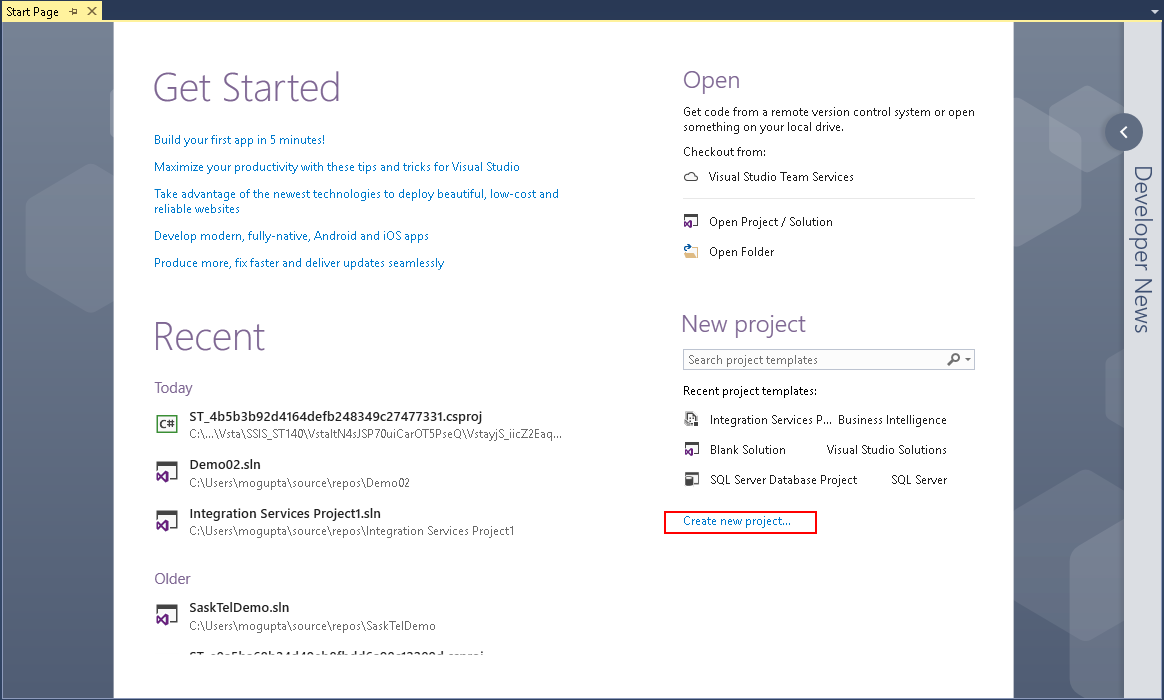
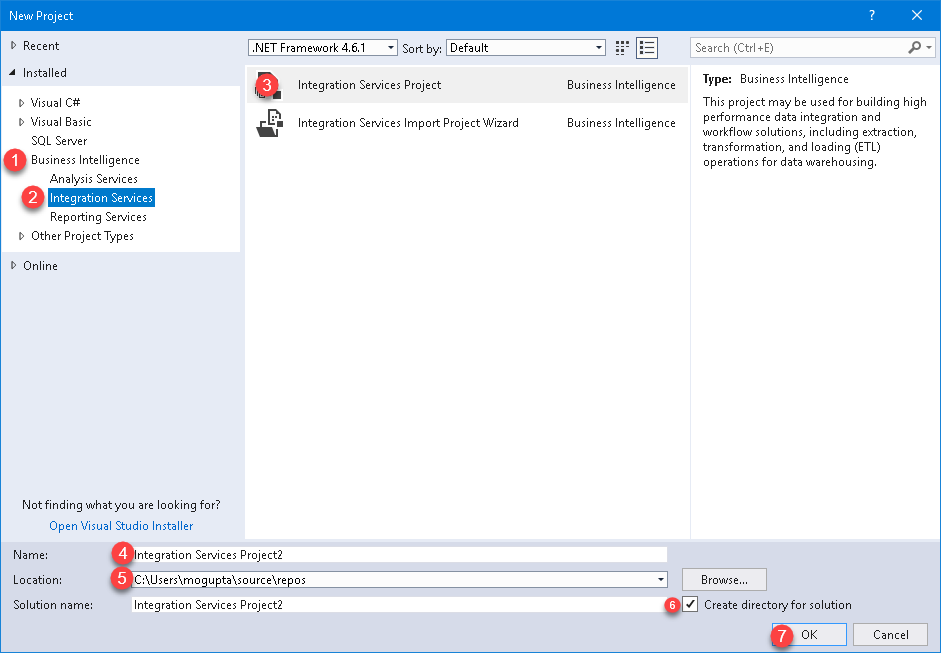
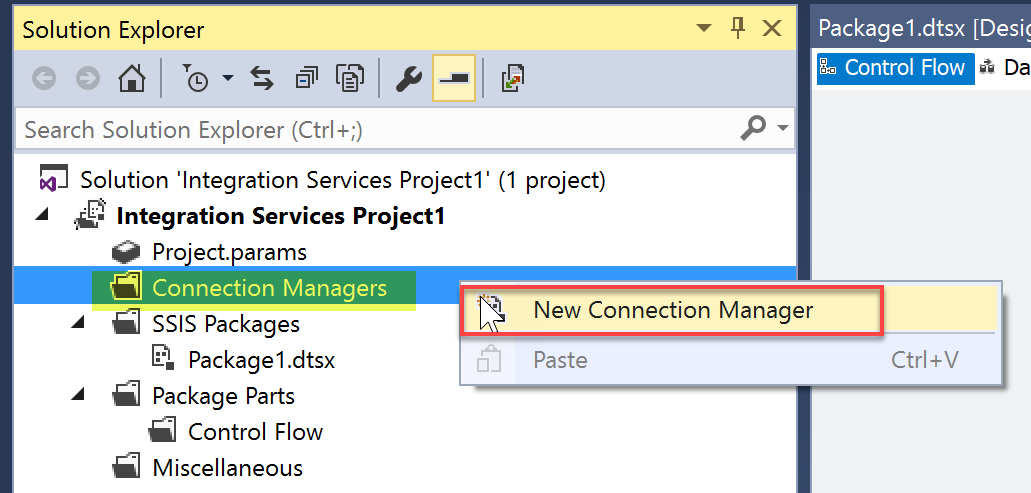
SQL Server Integration Services

# Module 03: CONTROL FLOW: SQL Execute TASK (Stored procedure)

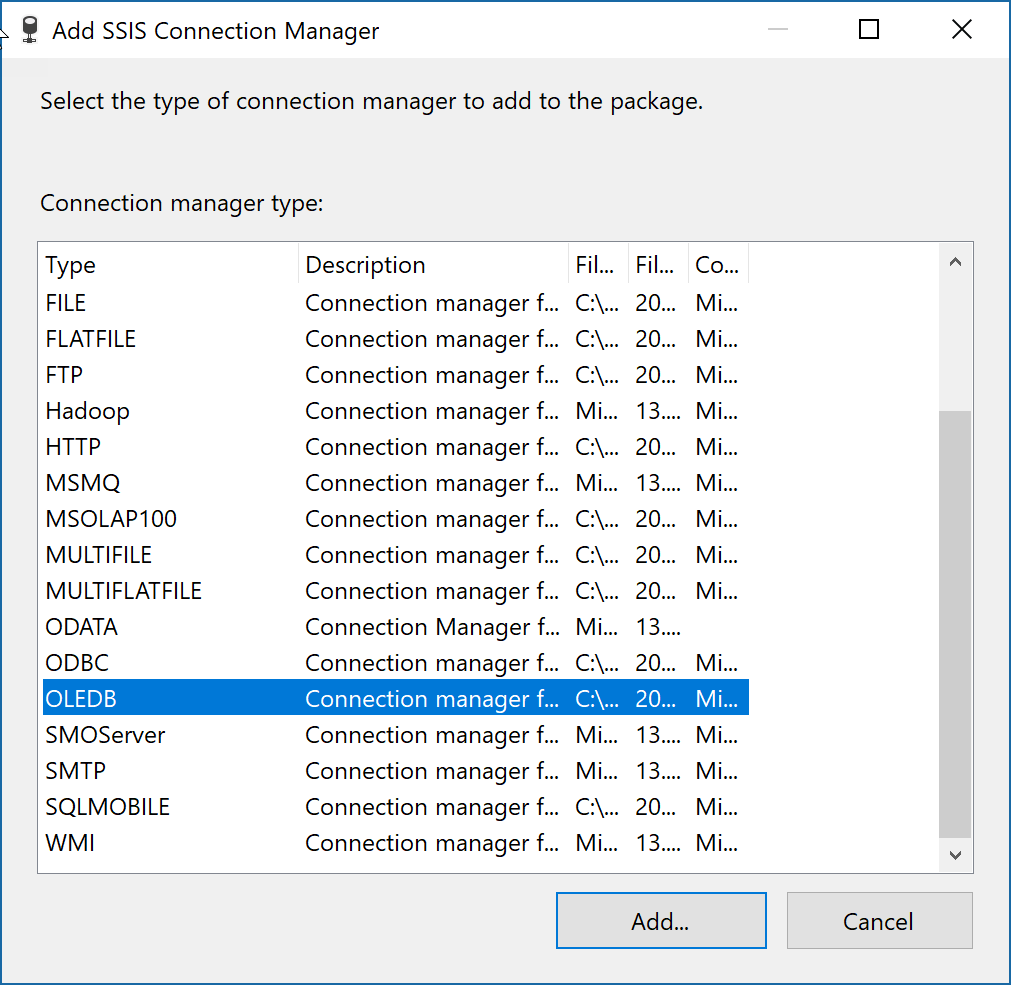
1. Launch SQL Server Data Tools (SSDT), under start menu look for Visual Studio 2017 (SSDT).
2. In the Start Page, click Create new project.  
   
3. In New Project dialog box select Business Intelligence > Integration Services > Integration Services Project. On the bottom enter in project name and location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



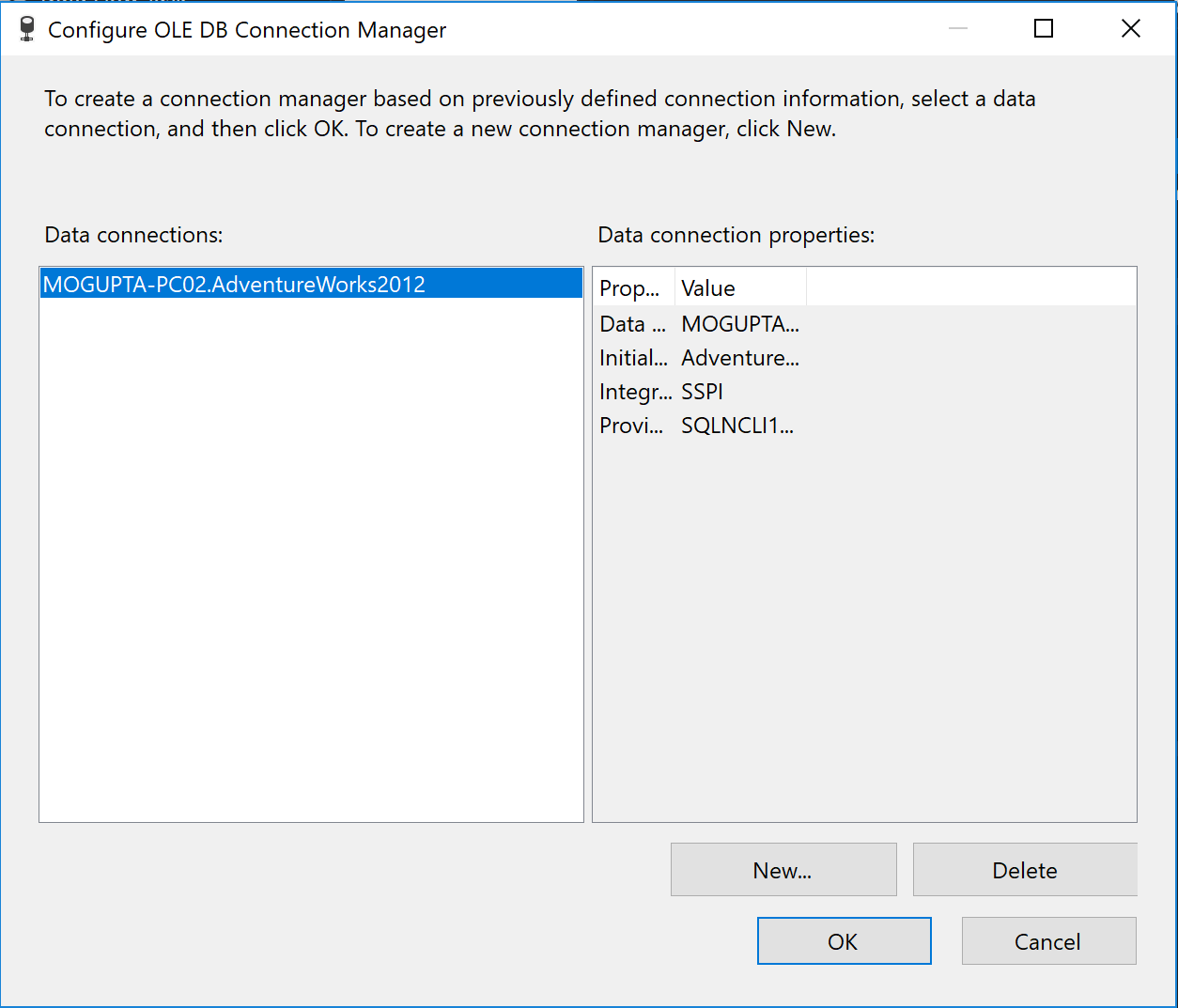
1. Create a Shared Connection Manager. In Solution Explorer, right-click connection manager and select New Connection Manager.

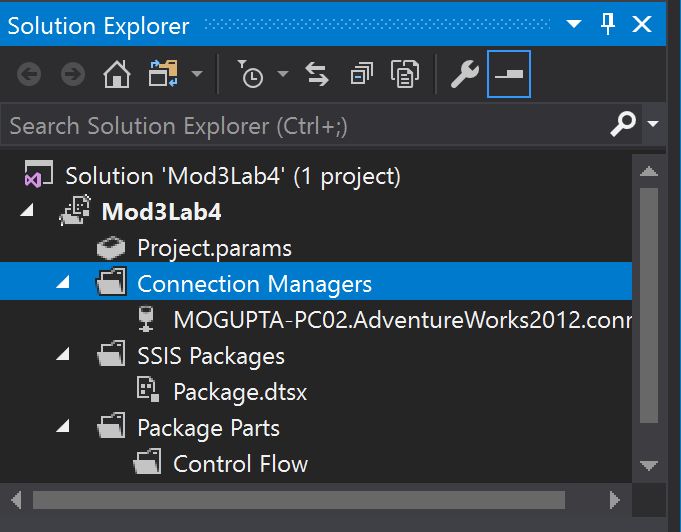


1. Select OLEDB and click Add.

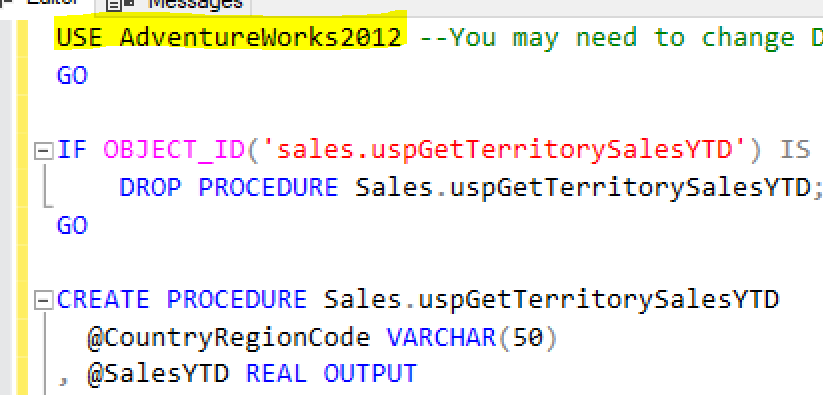


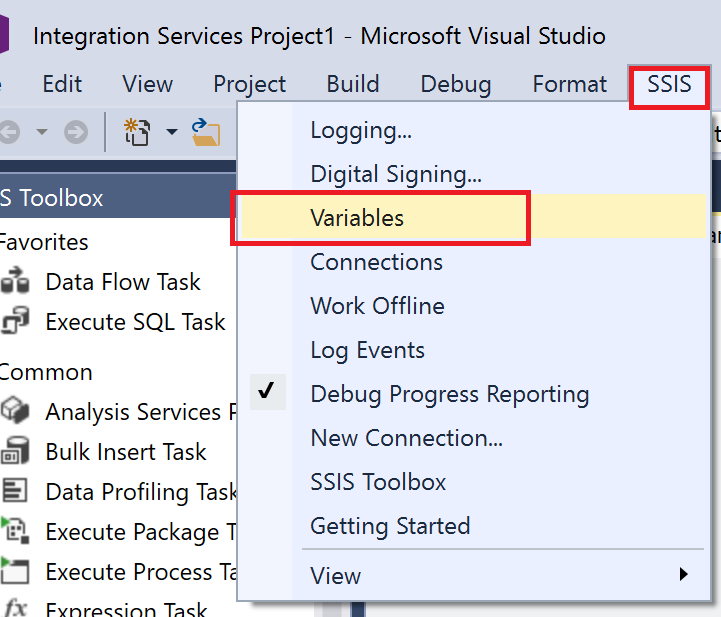
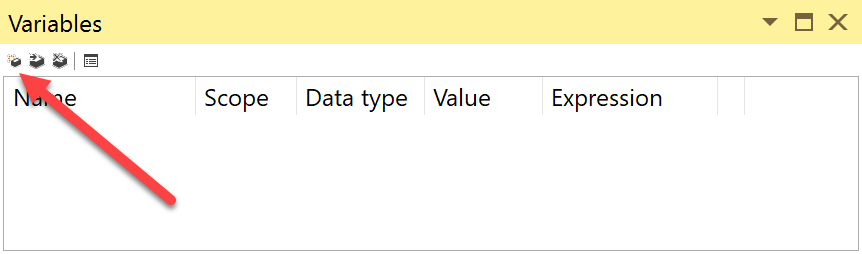
1. Select the connection Manager you have already created in previous Labs. And click OK.

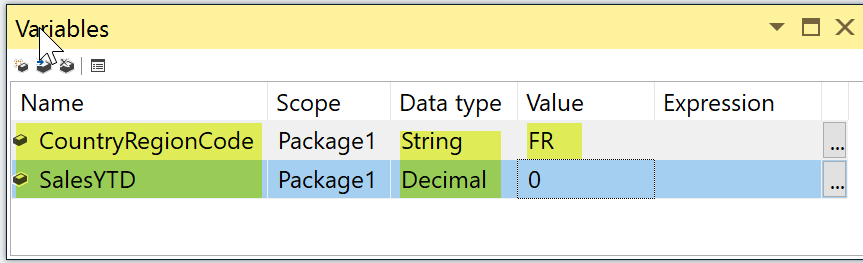


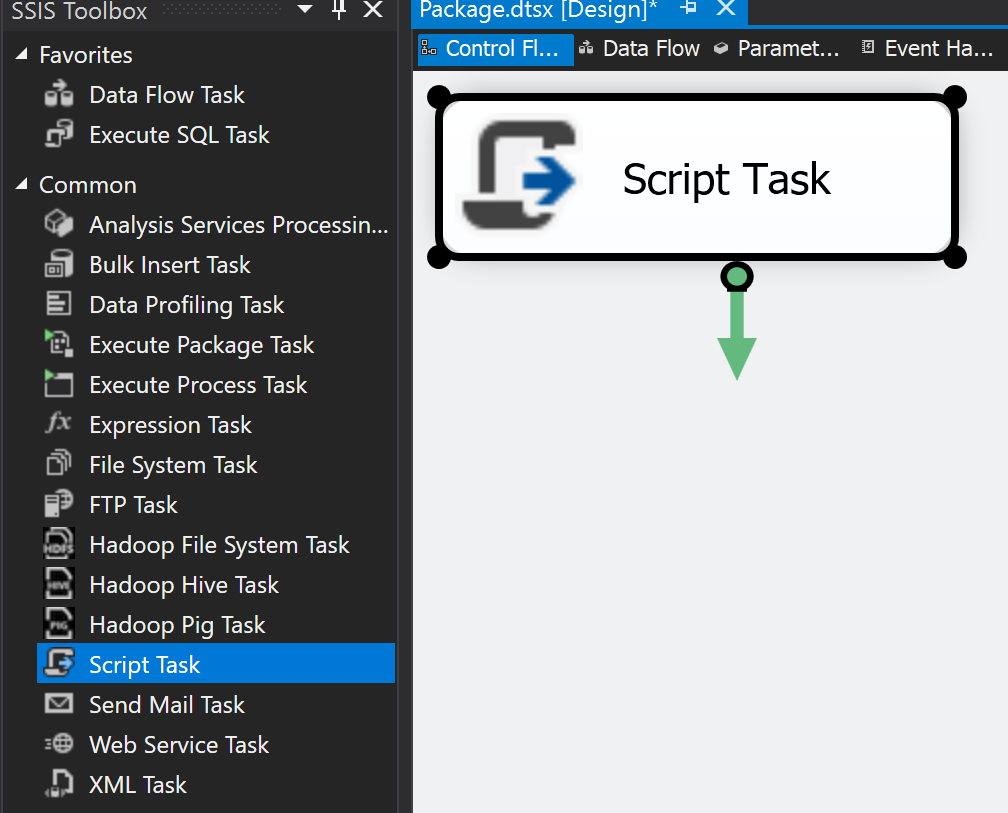


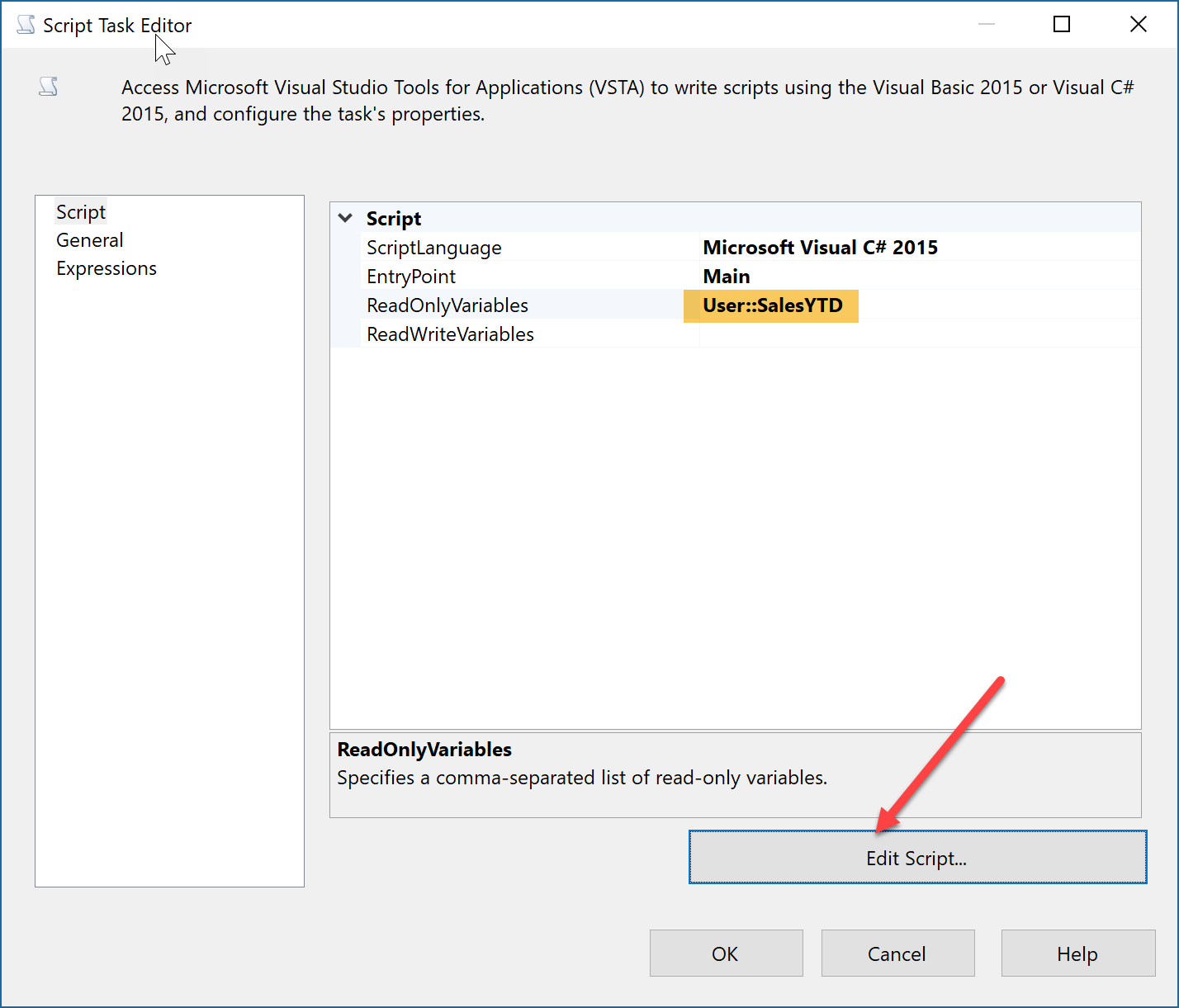
1. In SQL Server Management Studio (SSMS), open and run the Labs\Lab\_Files\Module03\Lab03\ StoredProcedure\_Sales\_uspGetTerritorySalesYTD.sql script to create a simple stored procedure we’ll need in later steps (make sure to update the database name).



1. Verify script runs correctly by testing it. Open and run the Labs\Lab\_Files\Module03\Lab03\ StoredProcedure\_Sales\_uspGetTerritorySalesYTD\_Test.sql script (make sure to update the database name).
2. Next, we are going to use Variables to call the stored procedure with input and output parameters, no result set is returned by the stored procedure and then you will display the result.
3. In the SSIS Menu select Variables.   
   
4. click on the New Variable icon  
   
5. Rename the Variable to CountryRegionCode of String Data Type, with a default value of FR. Rename second variable to SalesYTD of Decimal data type.

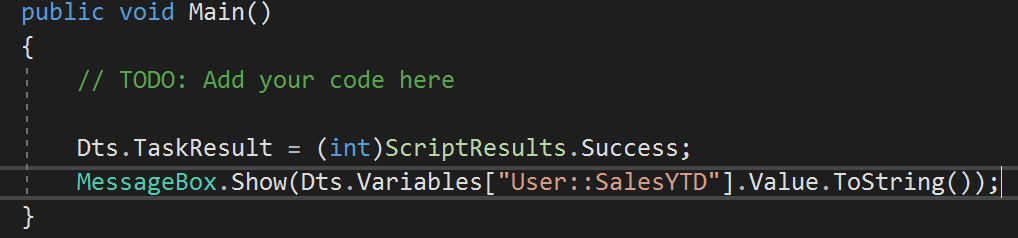


1. Double-click **Script Task** from SSIS Toolbox, to add onto the Control Flow canvas.   
   
2. Set Script Task’s ReadOnlyVariables property to User::SalesYTD in the Script Task Editor, then click on Edit Script.

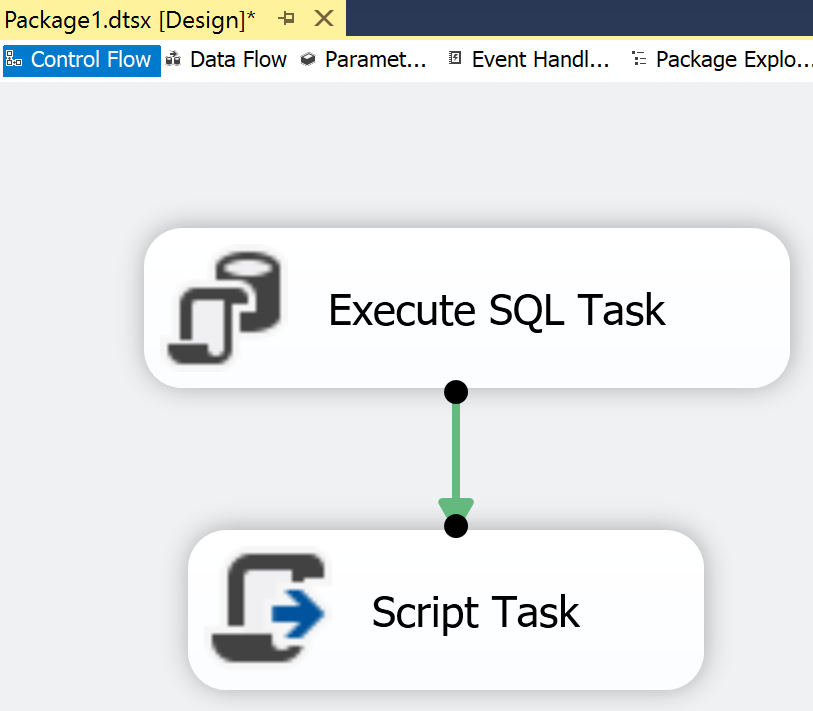


1. In the VstaProjects window, Add following code line into the main() block.

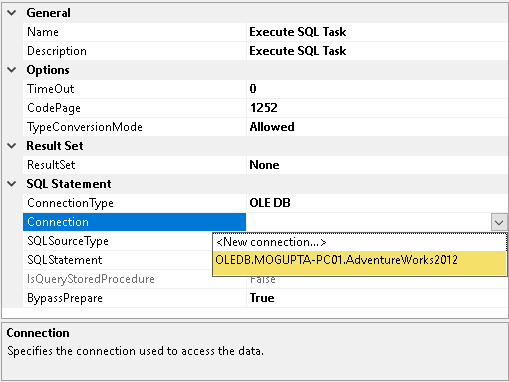
MessageBox.Show(Dts.Variables["User::SalesYTD"].Value.ToString());



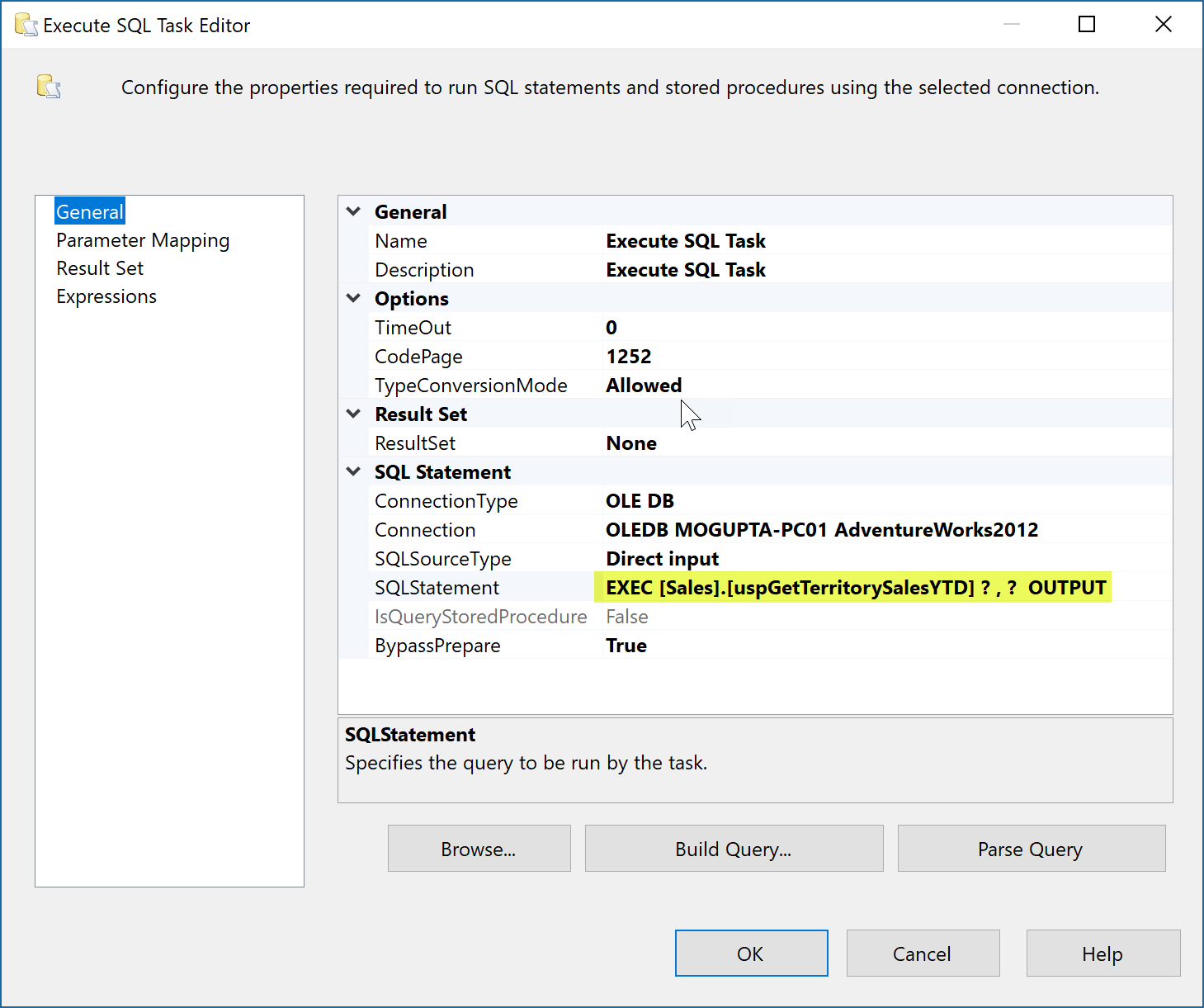
1. Close the VstaProjects (it will automatically save the code) , and then click on OK to close the Script Task Editor Dialog.
2. Double-click on **Execute SQL Task** from SSIS toolbox, to add onto Control Flow canvas.
3. Drag Green arrow from Execute SQL Task to Script Task.



1. Double click on the Execute SQL Task on the Control Flow canvas, and Set the Connection.

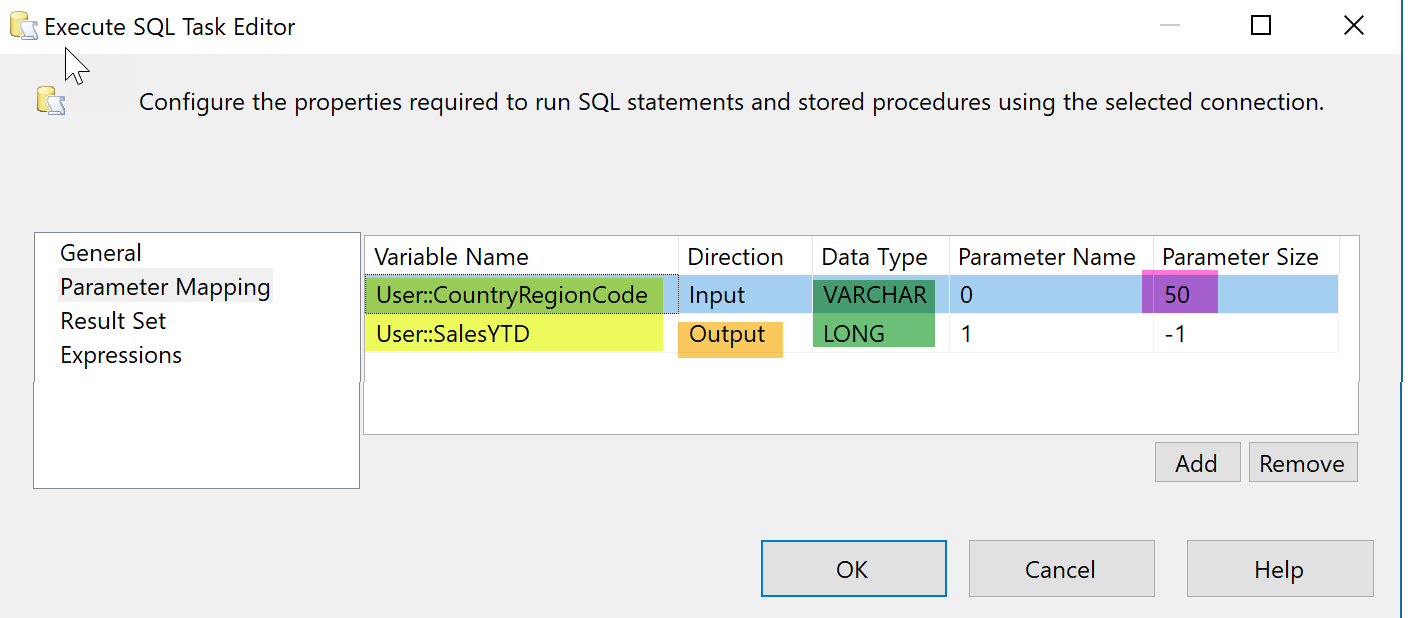


1. Enter the stored procedure all in the SQLStatement property

EXEC [Sales].[uspGetTerritorySalesYTD] ? , ? OUTPUT

*Note: Notice we are using the “****?****” (question mark) as markers for the parameters. Each driver has different requirements. Review the document,* [*Execute SQL Task – Parameter names and markers*](https://docs.microsoft.com/en-us/sql/integration-services/control-flow/execute-sql-task?view=sql-server-2017#Parameter_names_and_markers)*.*

1. Next, we must define how do stored procedure parameters map to variables in SSIS package. Click on the Parameter Mapping, and click Add to add two parameters as in the below screenshot.



*Note: The Parameter Name start from index 0 for OLDEB connections.*

1. Click OK to close the Execute SQL Task Editor dialog.
2. Execute the Package and notice the SaleYTD was successfully retrieved.

