

LEARNING ssis

Free unaffiliated eBook created from **Stack Overflow contributors.**

Table of Contents

About	1
Chapter 1: Getting started with ssis	2
Remarks	2
Examples	2
SSIS 2005 Installation	2
Chapter 2: Check if a file exists	. 4
Examples	4
Using loop control to run a dataflow task for every file	4
Steps to check if a file exist or not	5
Chapter 3: Convert datatype from Integer in YYYYMMDD format to Date	10
Examples	10
Using Built in conversion	10
Using Scripting Component	10
Chapter 4: Create a CSV file and write from SQL Server into that file	13
Introduction	13
Remarks	13
Examples	13
Steps to import data	13
Chapter 5: How to use variables inside a script component	20
Introduction	20
Parameters	20
Examples	20
Steps to achieve the objective	20
SSIS tasks required	20
Steps	20
Chapter 6: Load multiple CSV files of same format from a folder	29
Introduction	29
Parameters	29
Examples	29
Steps to load data	29

Chapter 7: Move file from one folder to another	
Examples	
File System Tasks in SSIS	35
Chapter 8: Read from a CSV file	38
Introduction	38
Examples	38
Read from a CSV file and insert data into a table	38
Chapter 9: Sorting incoming data, but send forward only a subset of rows	48
Examples	48
Using Sort and Conditional Split components	48
Credits	50

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: ssis

It is an unofficial and free ssis ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ssis.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with ssis

Remarks

This section provides an overview of what ssis is, and why a developer might want to use it.

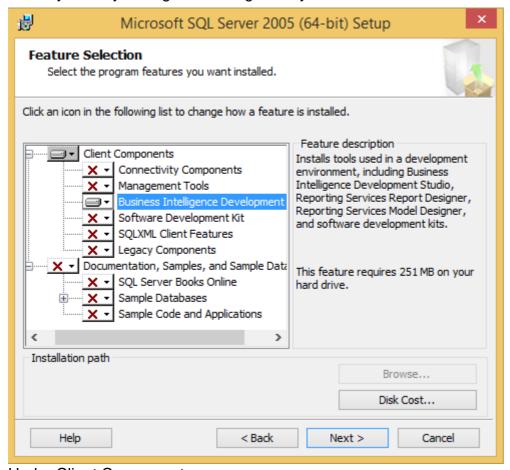
It should also mention any large subjects within ssis, and link out to the related topics. Since the Documentation for ssis is new, you may need to create initial versions of those related topics.

Examples

SSIS 2005 Installation

To get SSIS working for a SQL Server 2005 environment

- 1. Acquire SQL Server 2005 (x86 or 64 bit) images.
- 2. Mount the second disk and launch the installation wizard
- 3. "Next" your way through the dialogs until you see see this screen.



- 4. Under Client Components, ensure Business Intelligence Development Studio is selected
- 5. Continue clicking Next until the installation is complete.

You will now have BIDS (Business Intelligence Development Studio) on your machine with the correct project types so that you can create Integration Services, Analysis Services and Reporting

Services projects. BIDS uses a Visual Studio 2005 shell. Read Getting started with ssis online: https://riptutorial.com/ssis/topic/2557/getting-started-withssis

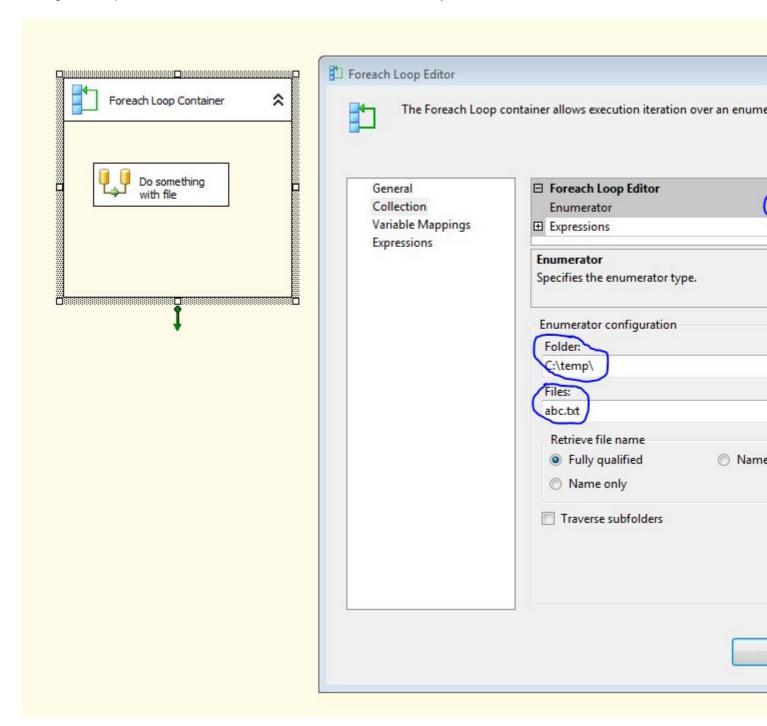
Chapter 2: Check if a file exists

Examples

Using loop control to run a dataflow task for every file.

If you want to check the existence of one file or do a couple of actions for every file in a folder you can use the Foreach Loop Container.

You give the path and the file mask and it will run it for every file it finds



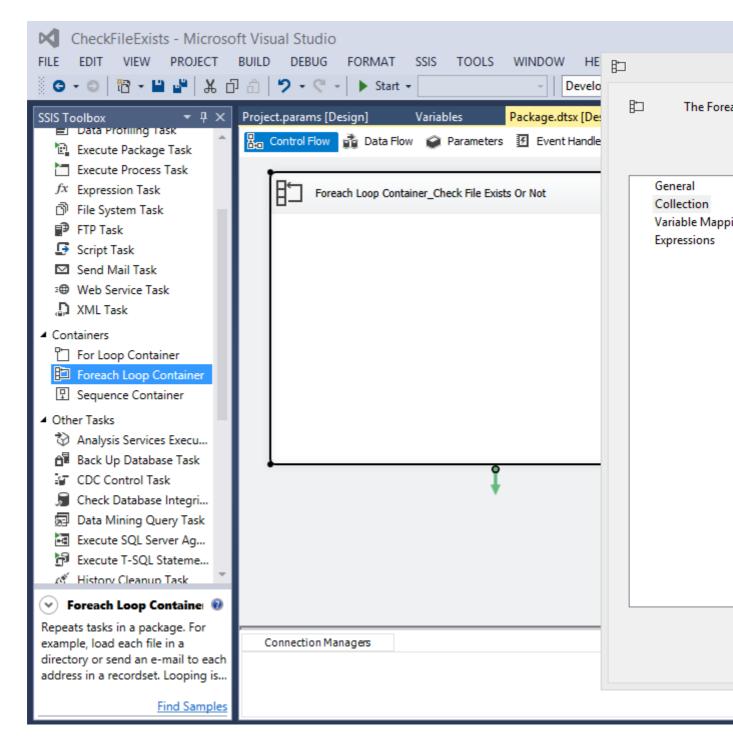
Steps to check if a file exist or not

To complete this objective following tasks are required.

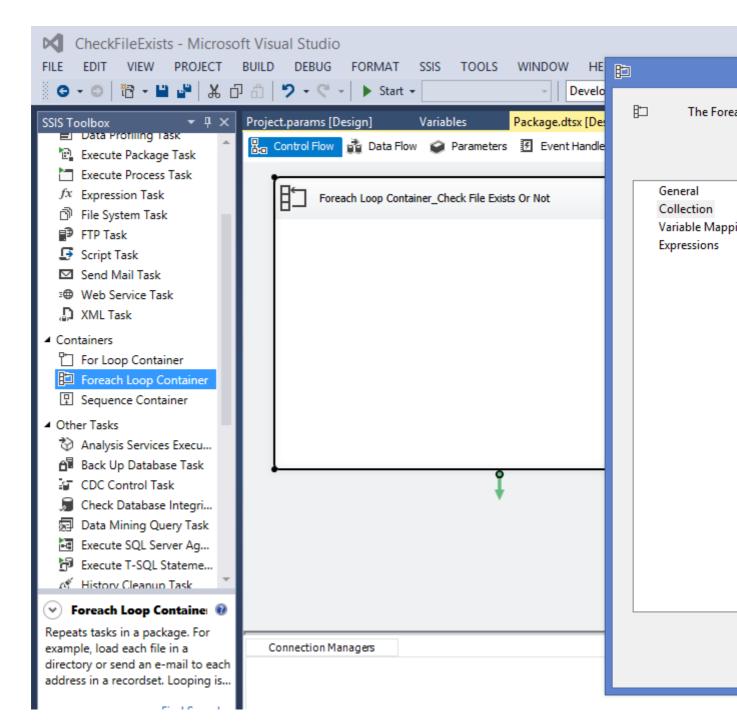
- 1. **Foreach Loop Container:** To iterate over a user configured directory for files.
- 2. Expression Task: To update a variable if file exists.

Steps

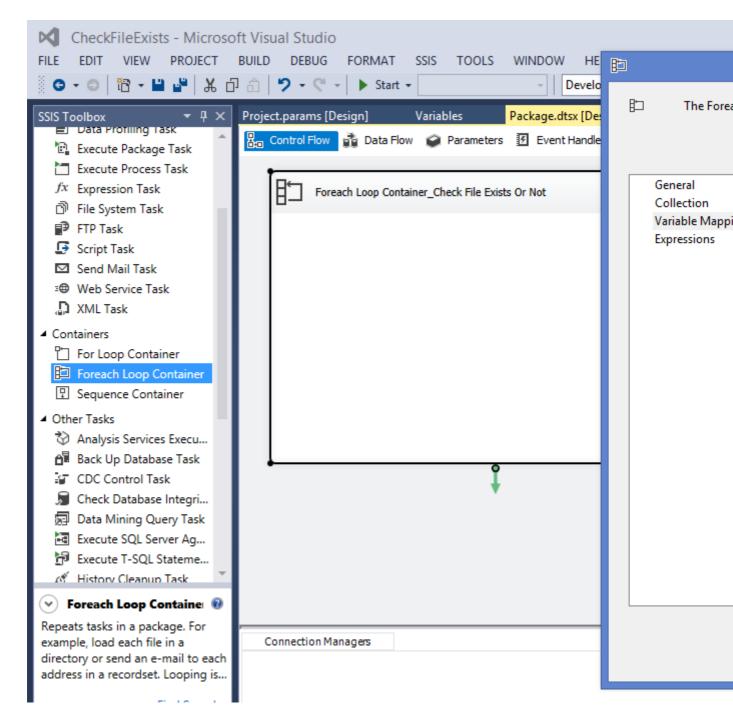
- 1. First goto **Solution Explorer** double click on **Project.params** and create a parameter *FolderPath* of type string, put value like E:\DataDir\SourceFiles.
- 2. Create user variables *FileNameFromFolder* (String), *FileToSearch* (String) assign value that you want to check and create a variable *IsFound* (Boolean).
- 3. Drag and drop a *Foreach Loop Container* from the SSIS Toolbox under Containers section.
- 4. Double click on the *Foreach Loop Container* on the left hand side of Foreach Loop Editor click on the **Collection**. On the right side set Enumerator as **Foreach File Enumerator**, now for the Expression click on the three dots which will open a Property Expression Editor. Select Directory as property and for expression select @[\$Project::FolderPath]. Click OK.



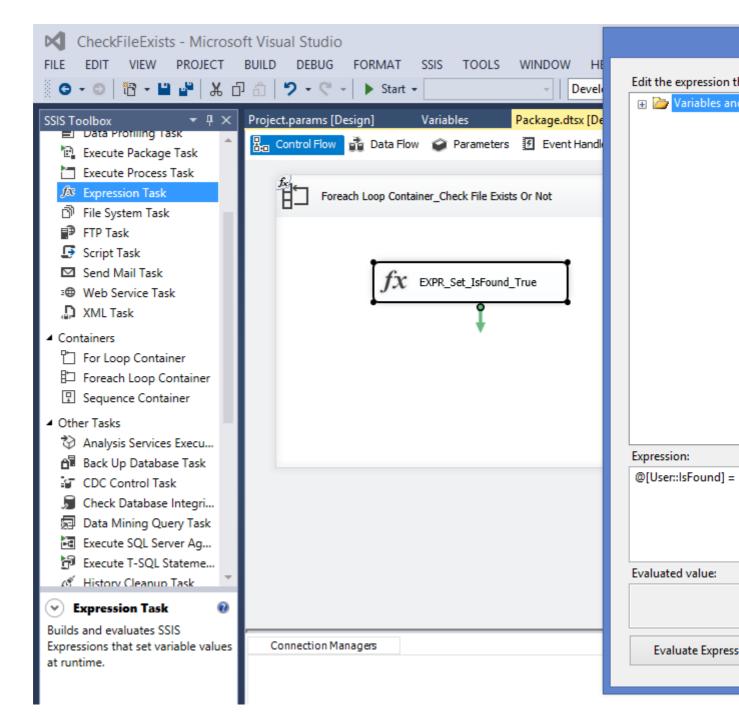
5. Now in Foreach Loop Editor for the value of Files set *.txt, for the value of *Retrieve file name* select Name only, normally we select Fully Qualified as it returns file name with the complete path. Check the *Traverse subfolders* if there can be more than one folder inside a folder.



6. On the left select **Variable Mappings**, on the right side select *User::FileNameFromFolder* which will automatically get Index as 0. The file names from the *FolderPath* will be assigned one by one to the *FileNameFromFolder* variable. Click OK.



- 7. Drag and drop a *Expression Task* inside the *Foreach Loop Container* from the SSIS toolbox present under the section **Common**.
- 8. Double click on the *Expression Task*, in the Expression Builder write following code. Click OK.
 - @[User::IsFound] = @[User::FileNameFromFolder] == @[User::FileToSearch] ? TRUE : FALSE



- 9. The Code above compares the file name that we want to check with the file name from the folder, if both are matched it sets *IsFound* to True (File Exists).
- 10. Now the value of IsFound can be used with precedence constraint according to need.

Read Check if a file exists online: https://riptutorial.com/ssis/topic/6617/check-if-a-file-exists

Chapter 3: Convert datatype from Integer in YYYYMMDD format to Date

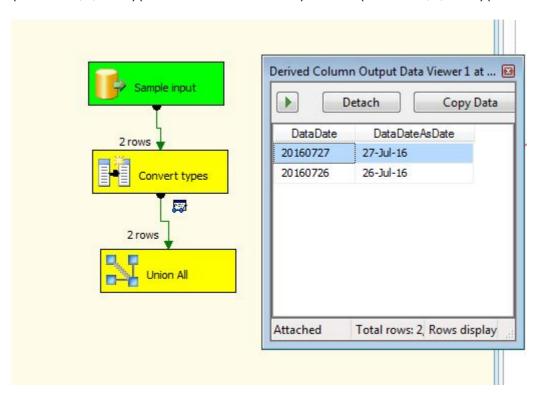
Examples

Using Built in conversion

Using Derived column we can prepare the input. We will provide yyyy-MM-dd to the final conversion:

- Year: (DT_STR,4,1252)(DataDate / 10000)
- Month: (DT STR,2,1252)(DataDate / 100 % 100)
- Day: (DT_STR,2,1252)(DataDate % 100)

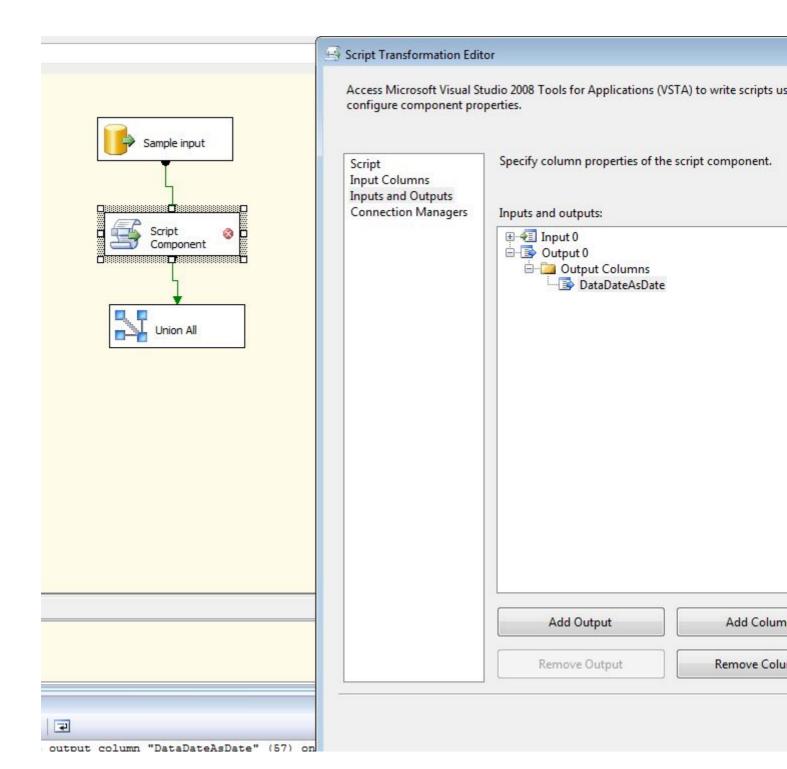
All together: (DT_DBDATE)((DT_STR,4,1252)(DataDate / 10000) + "-" + (DT_STR,2,1252)(DataDate / 100 % 100) + "-" + (DT_STR,2,1252)(DataDate % 100))



This is a faster solution than a scripting components, but less readable.

Using Scripting Component

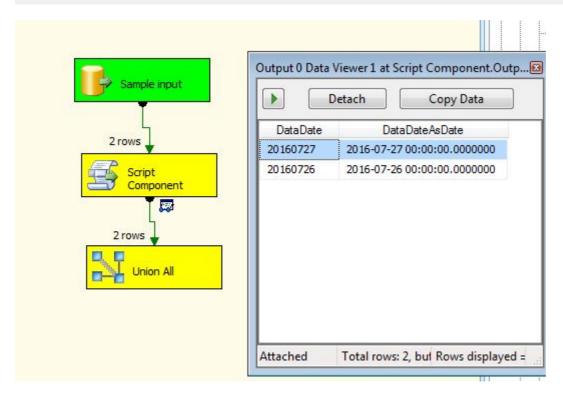
Using c# or vb.net code the conversion is even more simple. An output column is needed because we type can not be changed on the fly, alternative is adding an input column on forehand make it ReadWrite.



Next code will fill the new column.

```
public override void Input0_ProcessInputRow(Input0Buffer Row)
{
    if (Row.DataDate_IsNull)
        Row.DataDateAsDate_IsNull = true;
    else
    {
        DateTime tmp;
        if (DateTime.TryParseExact(Row.DataDate.ToString(), "yyyyMMdd", new
DateTimeFormatInfo(), System.Globalization.DateTimeStyles.None, out tmp))
        Row.DataDateAsDate = tmp;
    else
        // throw exception or return null
        Row.DataDateAsDate_IsNull = true;
```

)



Read Convert datatype from Integer in YYYYMMDD format to Date online: https://riptutorial.com/ssis/topic/4259/convert-datatype-from-integer-in-yyyymmdd-format-to-date

Chapter 4: Create a CSV file and write from SQL Server into that file

Introduction

The guide helps in understanding how to import data from the SQL server table to a CSV/txt file.

Remarks

- 1. Right click on the Data Flow Task and select property. **DefaultBufferMaxRows** and **DefaultBufferSize** properties can be changed to improve data load performance.
- 2. Multiple Data Flow Task can be executed in parallel for better performance.
- 3. Each Task has two flows success and failure. It is important to handle the failure flow to make the package more robust.
- 4. Inside the Data Flow Task right click on the blue arrow and select **Enable Data Viewer** to check data flow at run-time.
- 5. If any column is deleted in the source or the destination to check which column get deleted. Inside Data Flow right click on the blue arrow and select **Resolve References**, in the new window we can see Unmapped Output columns (left) and Unmapped Input Columns (right).

Examples

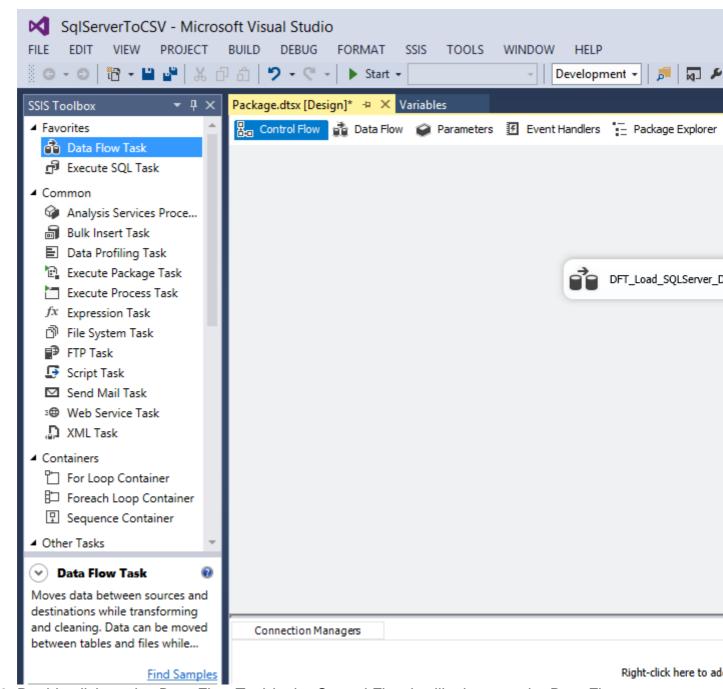
Steps to import data

Here is what is require to complete this Objective.

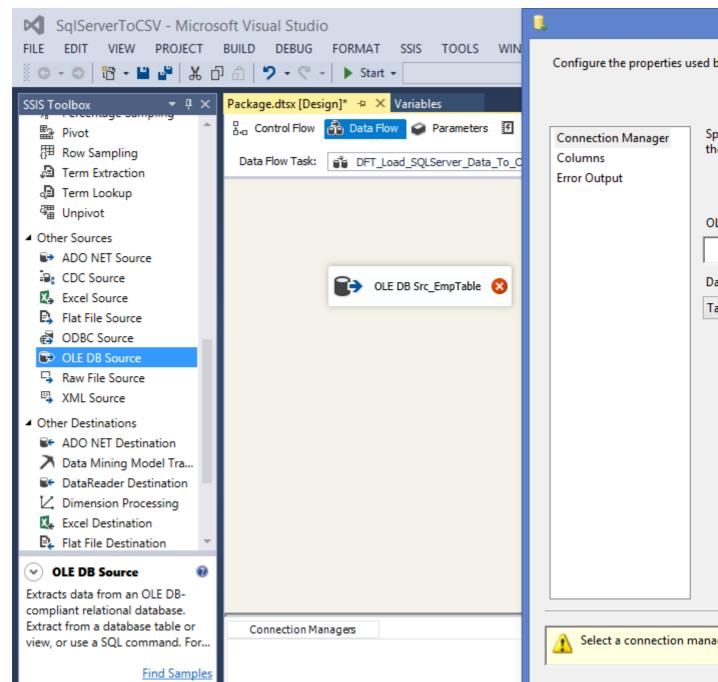
- 1. **Data Flow Task:** Inside this task we will perform data import.
- 2. OLE DB Source: To select the source of data i.e SQL server database table.
- 3. Flat File Destination: Destination in which we want to load the data.

Steps

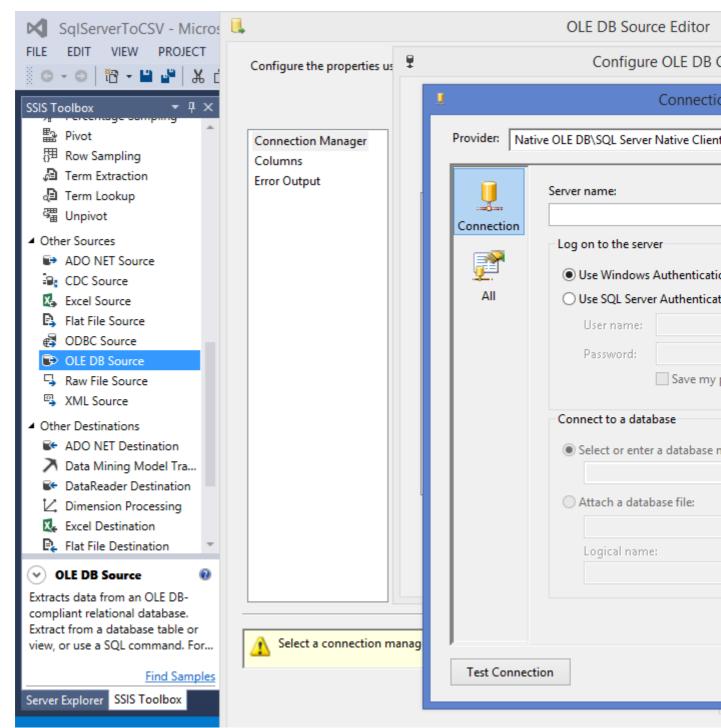
1. Drag and drop a *Data Flow Task* from the SSIS toolbox from the Favorites section.



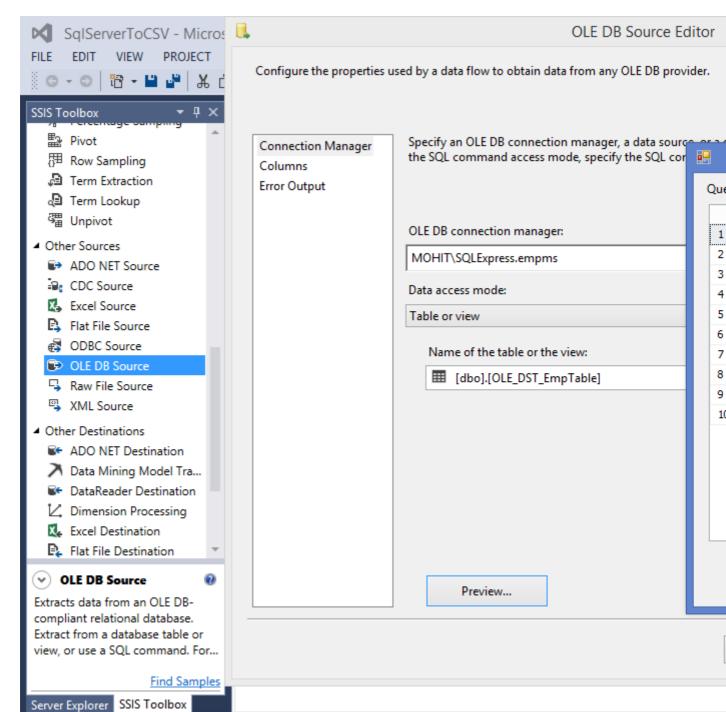
- 2. Double click on the Data Flow Task in the Control Flow it will take us to the Data Flow.
- 3. Drag and drop a *OLE DB Source*, by default a cross will appear on it, it means it is not configured with a connection. Double click on the *OLE DB Source* task, click on the New.



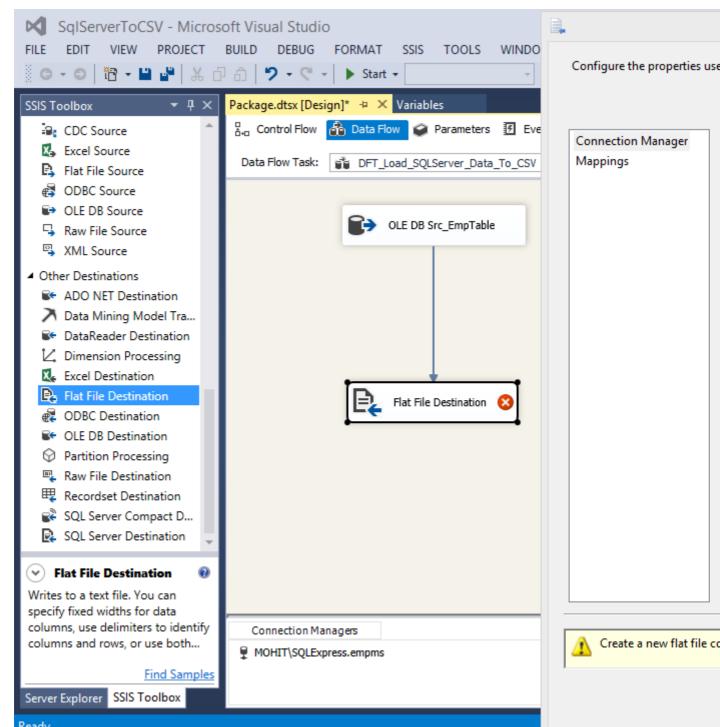
4. On the Configure OLE DB Connection Manager window click New. Now in the Connection Manager window select the Server Name to which you want to connect. Select Windows Authentication if your server is on your machine otherwise Use SQL Server Authentication and enter user name and password. Click on the **Test Connection** at the buttom left to check the validity of the credential entered. Click OK and then again OK.



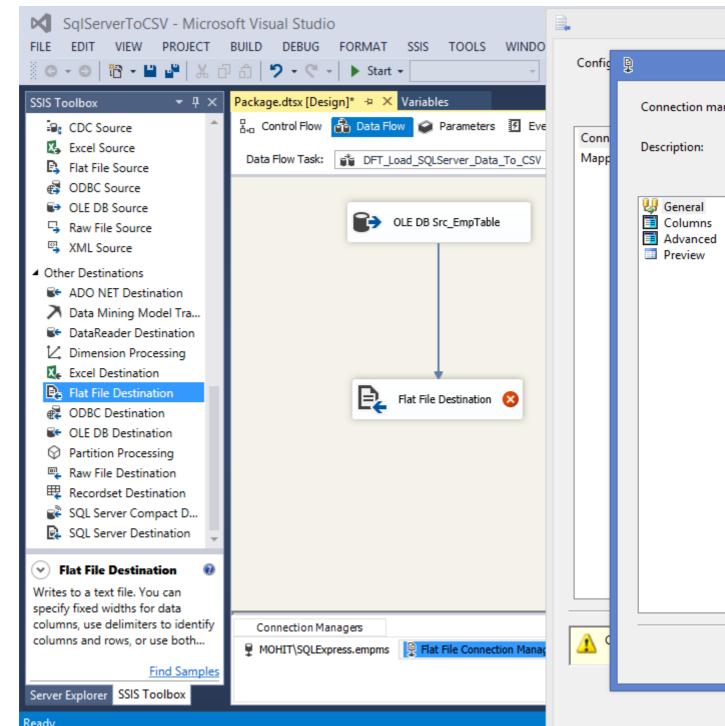
5. In the OLE DB Source Editor select name of the table or the view and click on the Preview to check the data. Click Close then OK.



- 6. Drag and drop a *Flat File Destination* task from the SSIS toolbox under the section other destinations. Connect the *OLE DB Source* to the *Flat File Destination*.
- 7. Double click on the *Flat File Destination*, click on New it will open Flat File Format window. Select **Delimited** if you want to specify the separator, text qualifier, end of line etc. Click OK.



- 8. In the Flat File Connection Manager Editor click Browse button, select the path for file and enter file name click Open. Even though we have not selected any file we have just entered the file name the file will get created.
- 9. Now select the Code Page, Text Qualifier etc. Remember to tick the checkbox **column name in the first data row**. On the left side select Columns here you can specify the data separator like comma or a pipe (|). Click OK.



- 10. In the Flat File Destination Editor overwrite data in the file is selected, update it according to requirement. On the left select Mappings and check if columns are mapped correctly. Click OK.
- 11. In the Solution Explorer right click on the package name and execute it to check.

Read Create a CSV file and write from SQL Server into that file online: https://riptutorial.com/ssis/topic/9839/create-a-csv-file-and-write-from-sql-server-into-that-file

Chapter 5: How to use variables inside a script component

Introduction

This post provides steps to use variables (User Variable, Package Parameter and Project Parameter) in the script component and viewing the updated value using Breakpoint and Watch window.

Parameters

Parameter	Details
UserVar	It is like a local variable used inside a package. Its value can be read and modified in script task
Result	It is a local variable which will hold the concatenated result. Its value can be read and modified in script task
PackageVar	It is package parameter, which can be shared between packages. Its value is read only inside script component
ProjectParm	It is project parameter, which is available after deployment as a configuration. Its value is read only inside the script component

Examples

Steps to achieve the objective.

SSIS tasks required.

- 1. Data Flow Task: As the script component is only available in the data flow.
- 2. **Script Component:** Inside this we will use variables and play with there values.

Steps

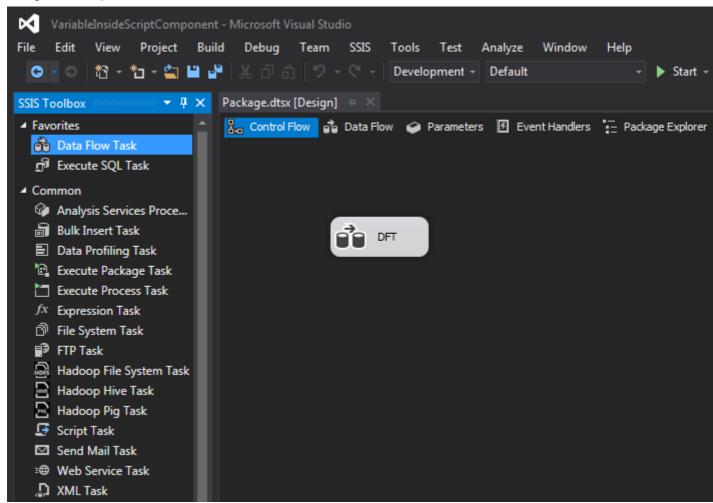
There are two methods to access variables inside of script component

First Method - Using this. Variables

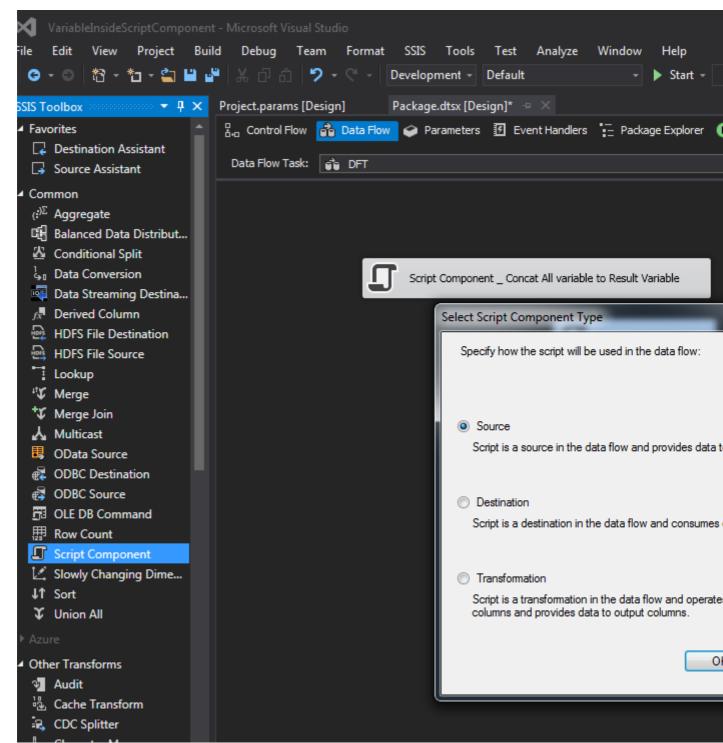
1. Create two user variable Result (String), UserVar (String value: UserVar), also create a package parameter PackageVar (String value: PackageVariable) and a project parameter

ProjectParam (String value: ProjectParameter).

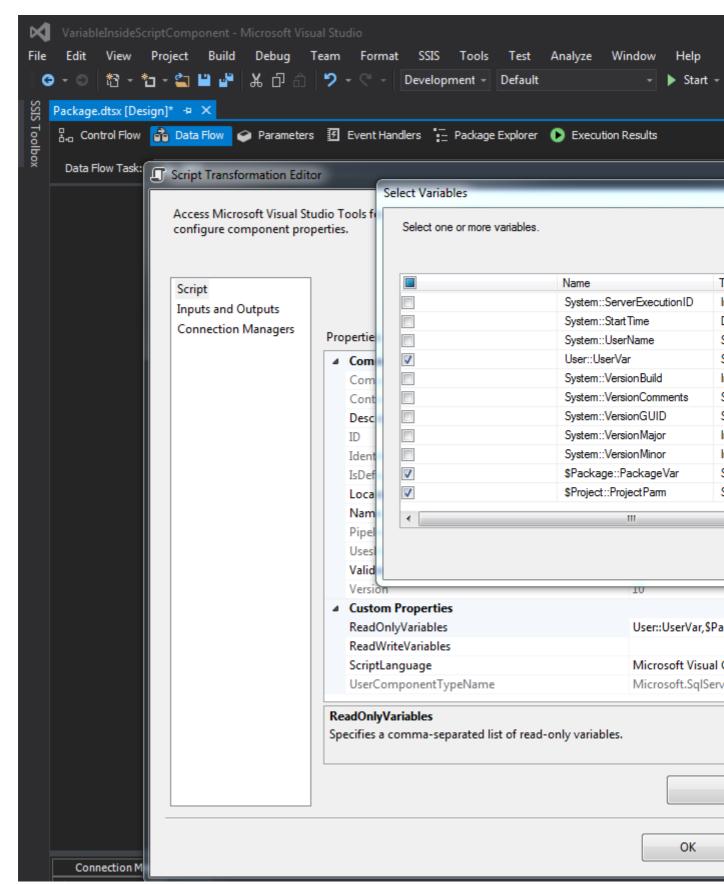
2. Drag and drop a Data Flow Task from the SSIS Toolbox under favorites section.



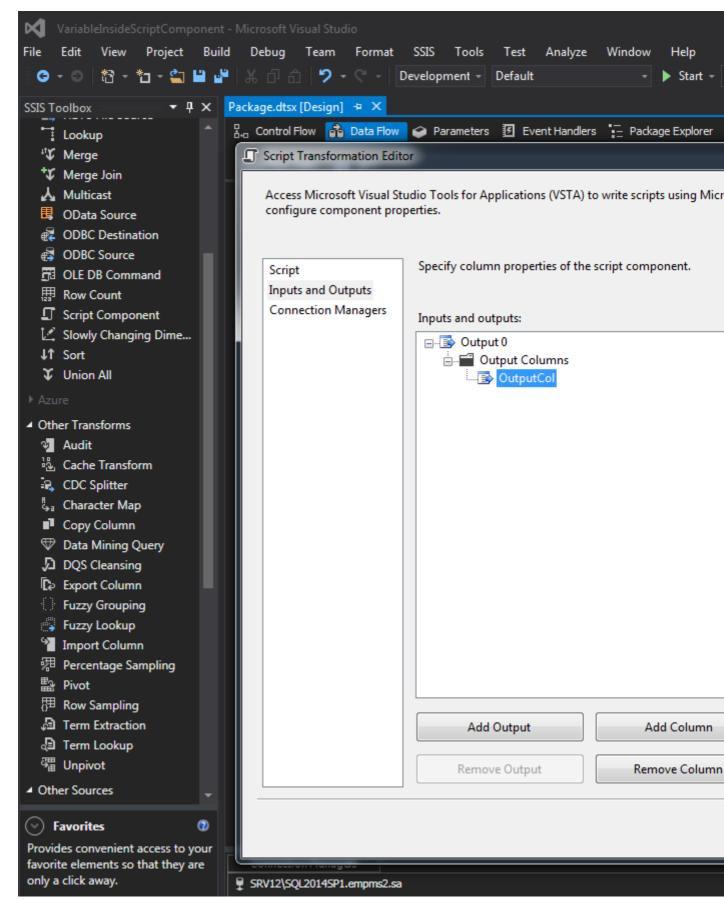
3. Double click on the *Data flow task* which will take you into the Data Flow. Now from the SSIS Toolbox drag and drop *Script Component* present under *Common* section. It will prompt with a window having three choices Source, Destination and Transformation. Select Source and click OK.



4. Double click on the *Script component*, on the right hand side for **ReadOnlyVariables** click on the three dots it will open a *Select Variable* window. Now select User::UserVar, \$Package::PackageVar and \$Project::ProjectParm. Click OK. Similarly Click on the three dots corresponding to the **ReadWriteVariables** and select User::Result. Click OK.



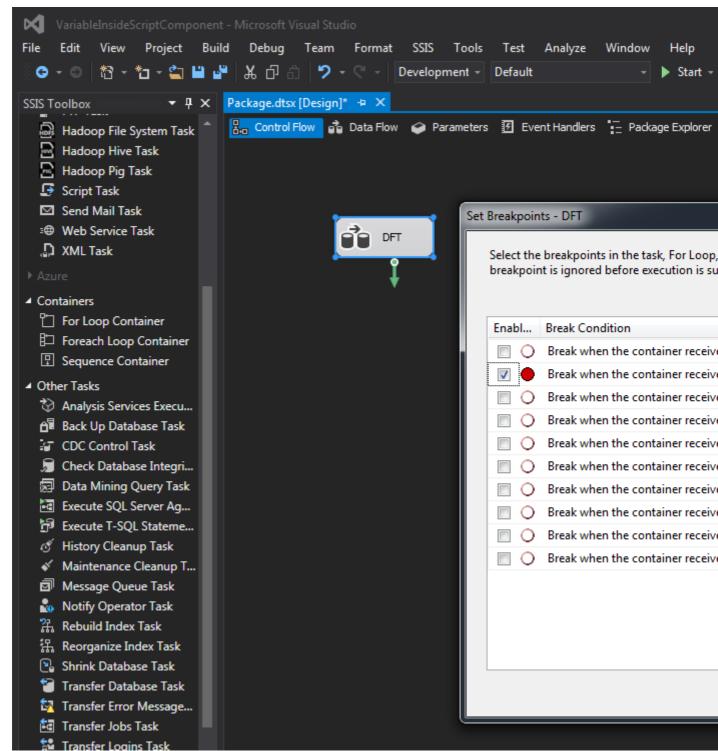
5. On the left hand side of Script Transformation Editor select *Inputs and Outputs*, On center Expand **Output O** -> expand Output Columns at bottom Click *Add Column*. When we use script component as Source, it is required to have output column, that is why we have created this output column.



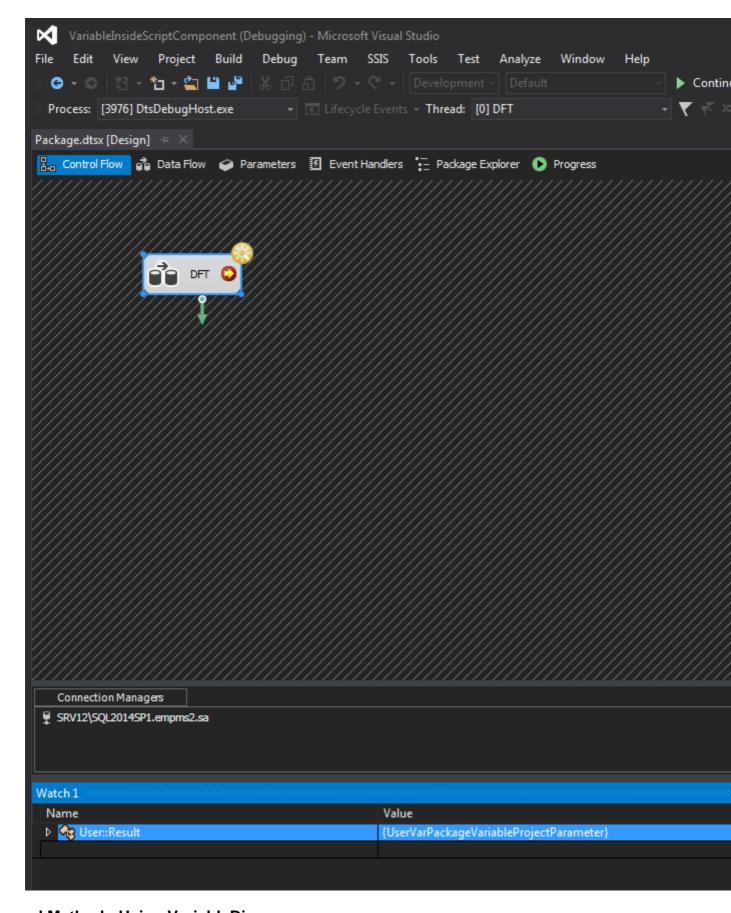
6. On the left Click on the *Script*, on the right bottom of the editor click *Edit Scripts*, it will open a new window, in this window find **PostExecute()** method and write this.Variables.Result = this.Variables.UserVar + this.Variables.PackageVar + this.Variables.ProjectParm; through this.Variables we are accessing variable. Click Ctrl+S to save and close the window. Click OK.

```
×
    VstaProjects - Microsoft Visual Studio
File
                          Build
                                 Debug
                                                Tools
                                                                       Window
                                                                                Help
           View
                  Project
                                         Team
                                                        Test
                                                              Analyze
                                           り・
 G - 🖯 | 👸 - 🐮 - 🖆 💾 🛂
                                * 🗗 🗈
   main.cs 🗢 X
   C# SC_c9f3271fc4d742c2beb9063b264941f1
                                                               ScriptMain **
                /// </summary>
                [Microsoft.SqlServer.Dts.Pipeline.SSISScriptComponentEntryPointAttribute]
              □public class ScriptMain : UserComponent
               |{
                    Help: Using Integration Services variables and parameters
                    Help: Using Integration Services Connnection Managers
                    Help: Firing Integration Services Events
                    /// </summary>
                    1 reference
                    public override void PreExecute()
                        base.PreExecute();
                    /// <summary>
              ፅ
                    public override void PostExecute()
                        base.PostExecute();
                         * Add your code here
       103 😨
                        this.Variables.Result = this.Variables.UserVar + this.Variables.PackageVar
                    public override void CreateNewOutputRows()
              ൎ
                    {
                          Add rows by calling the AddRow method on the member variable named "<Outp
```

7. Go to the Control Flow and right click on the *Data Flow Task* and select **Edit Breakpoints**. Now in the new *Set Breakpoints* window *Select Break when the container receives the OnPostExecute event*. Click OK.



8. Now in the *Solution Explorer* right click on the package name and click Execute Package. From the menu bar click on the Debug -> Windows -> Watch -> Watch 1. Now at the bottom Watch window will be visible. Under *Name* type User::Result and click enter. Under *value* concatenated values {UserVarPackageVariableProjectParameter} can be seen.



Second Method - Using VariableDispenser

When the variable dispenser is used adding the variables to the <code>ReadOnlyVariables</code> and <code>ReadWriteVariables</code> is not needed. You can use the following codes to Read and write variables values: (Code is for ssis 2008)

Write

```
private void WriteVariable(String varName, Object varValue)
{

IDTSVariables100 vars = null;
VariableDispenser.LockForWrite(varName);
VariableDispenser.GetVariables(out vars);
vars[varName].Value = varValue;
vars.Unlock();
}
```

Read

```
private Object ReadVariable(String varName)
{
    Object varValue;
    IDTSVariables100 vars = null;
    VariableDispenser.LockForRead(varName);
    VariableDispenser.GetVariables(out vars);
    varValue = vars[varName].Value;
    vars.Unlock();
    return varValue;
}
```

Read How to use variables inside a script component online:

https://riptutorial.com/ssis/topic/9889/how-to-use-variables-inside-a-script-component

Chapter 6: Load multiple CSV files of same format from a folder

Introduction

In this guide you can find out steps to load multiple CSV/TXT files from a folder to the database table.

Parameters

Parameter/Vaiable	Details
SourceFolder	It is a read only project parameter available and configurable at the deployment. Example of project parameter are Connection Strings, passwords, port no, users etc
CompleteSourceFilePath	It is read write user variable available only inside the package like local variables in programming languages

Examples

Steps to load data

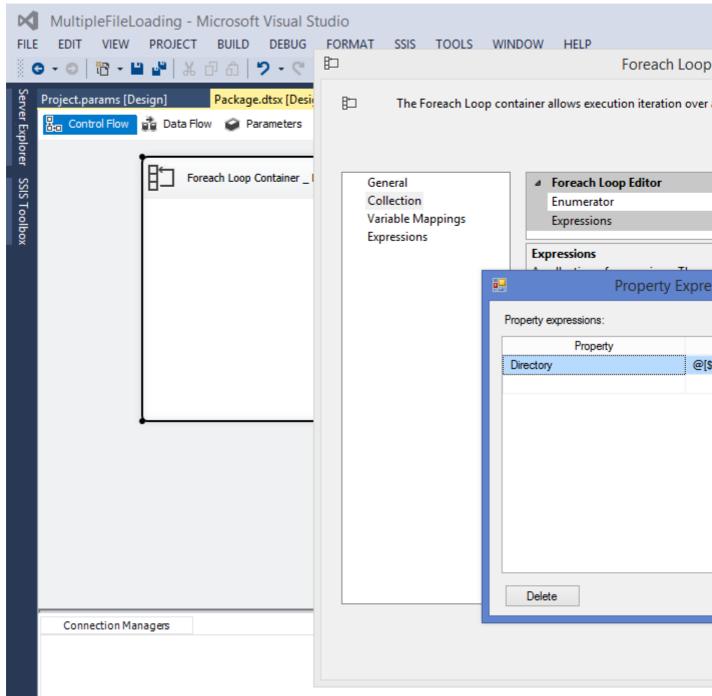
To achieve this objective what we need is

- 1. Foreach Loop Container: To Iterate over a directory to pick files.
- 2. **Data Flow Task:** To load data from the CSV (Flat File Source) to the database table (OLE DB Destination).
- 3. Flat File Source: For text or csv files.
- 4. **OLE DB Destination:** To select the destination table which we want to populate with.

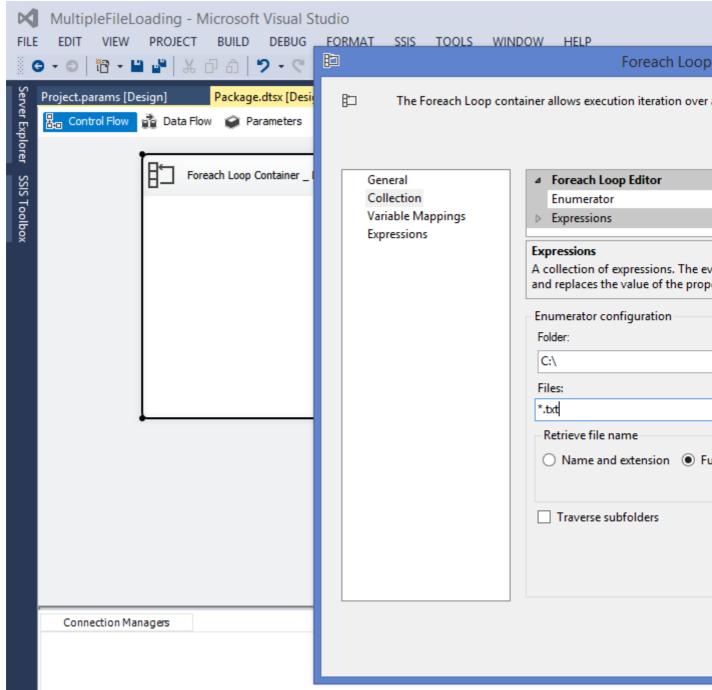
Steps

- 1. First Drag and drop a *Foreach Loop Container* from the **container** section of the SSIS toolbox.
- 2. Now double click on the **Project.params** in the solution Explorer and create a variable SourceFolder as string. In the value field type the path from which you want to pick the files. We are creating this path as a project parameter so that it can be configured after deployment.
- 3. Create a user variable by clicking on the **Variables** icon on the right and create CompleteSourceFilePath variable of type string. This variable will hold the value returned from the Foreach loop container.
- 4. Now Double click on the *Foreach loop Container* and select Collection on left hand side. On

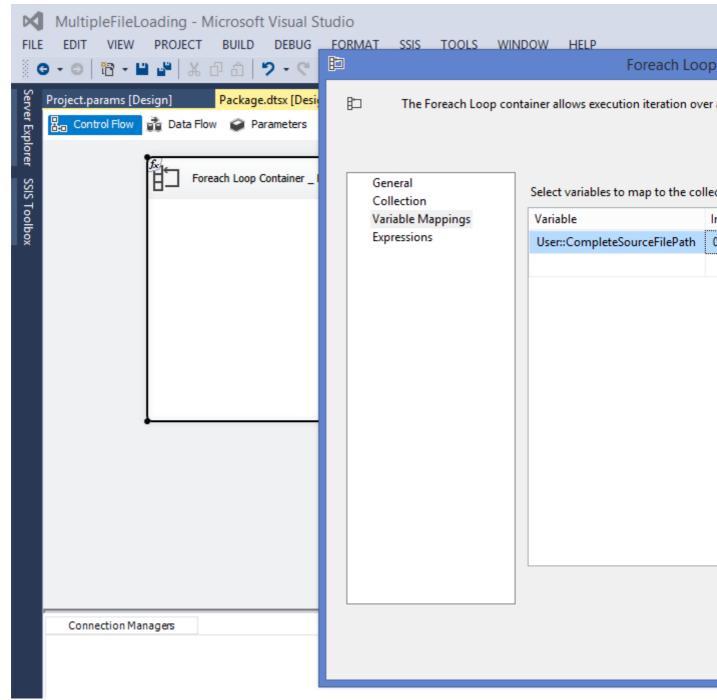
the right hand side select **Foreach File Enumerator**. Now for Expression click on the three dots on the right which will open a property editor select **Directory** in the property section and select @[\$Project:SourceFolder] as its value. Click Ok.



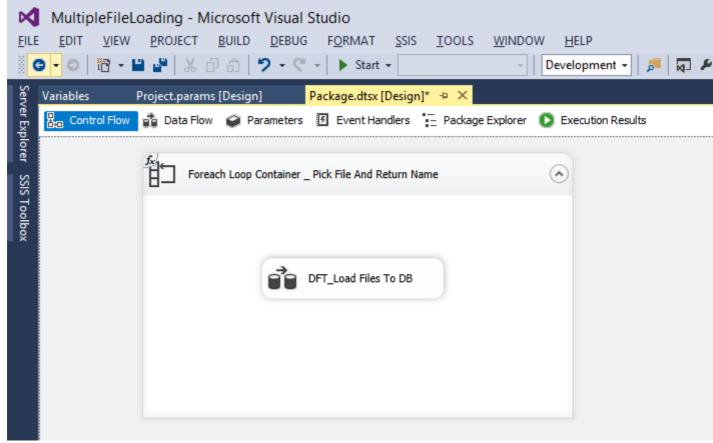
5. In Foreach Loop Editor window for Files enter *.txt or *.csv whatever file extension is required.



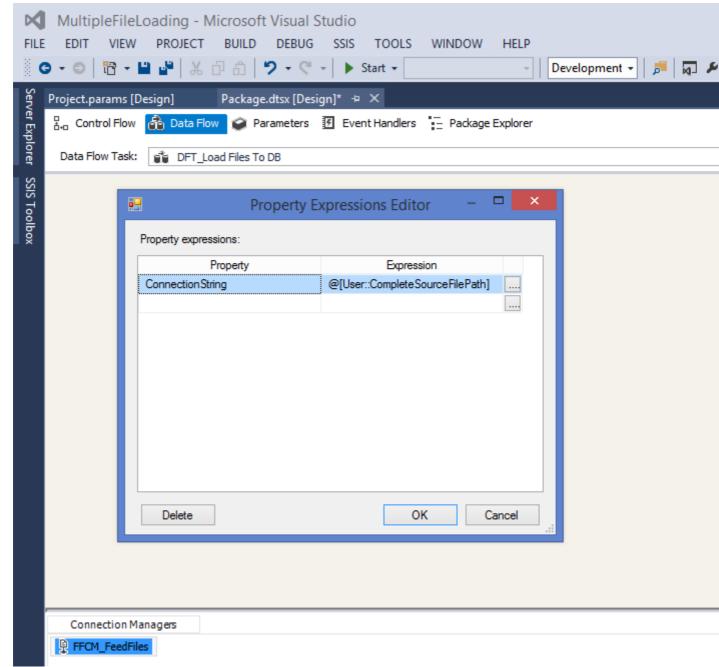
6. On the left hand side of the Foreach Loop Editor select **Variable Mappings**, on the right select *User::CompleteSourceFilePath* which will automatically assigned Index 0. Click OK.



7. From the SSIS toolbox drag and drop *Data Flow Task* from the favorites section inside the *Foreach Loop Container*. Each file name returned by the *Foreach loop container* in CompleteSourceFilePath variable will be used in the *data flow task*.



- 8. Now double click on the Data Flow task which will take us into the Data flow. Drag and drop a *Flat File Source* from the other Source section of the toolbox.
- 9. At the bottom of the screen in **Connection Managers** section right click and then select New *Flat File Connections*. Click on the Browse button and select one of the file that you want to process, set other properties like Text qualifier (like double quote). Click OK.
- 10. Click on the new Flat file connection created in the connection Manager section and go the Properties window. Find the Expressions property and click on the three dots on the right. In the Property section select ConnectionString and in the Expression select the @[User::CompleteSourceFilePath] variable. Click OK.



- 11. Select a *OLE DB Destination* (according to the database) and configure it to the table that you want to load.
- 12. Right click on the package name (solution explorer) and click Execute package to test the package.

Read Load multiple CSV files of same format from a folder online:

https://riptutorial.com/ssis/topic/9838/load-multiple-csv-files-of-same-format-from-a-folder

Chapter 7: Move file from one folder to another

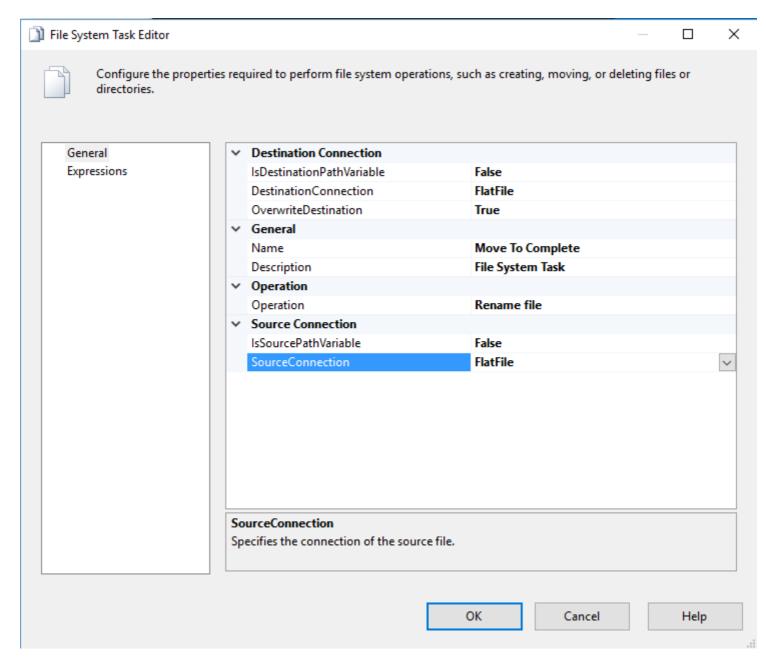
Examples

File System Tasks in SSIS

From the Control Flow tab in your SSIS package, look in the SSIS Toolbox in the common section for the File System Task, drag this onto your design surface where you want the file move to happen in your package.



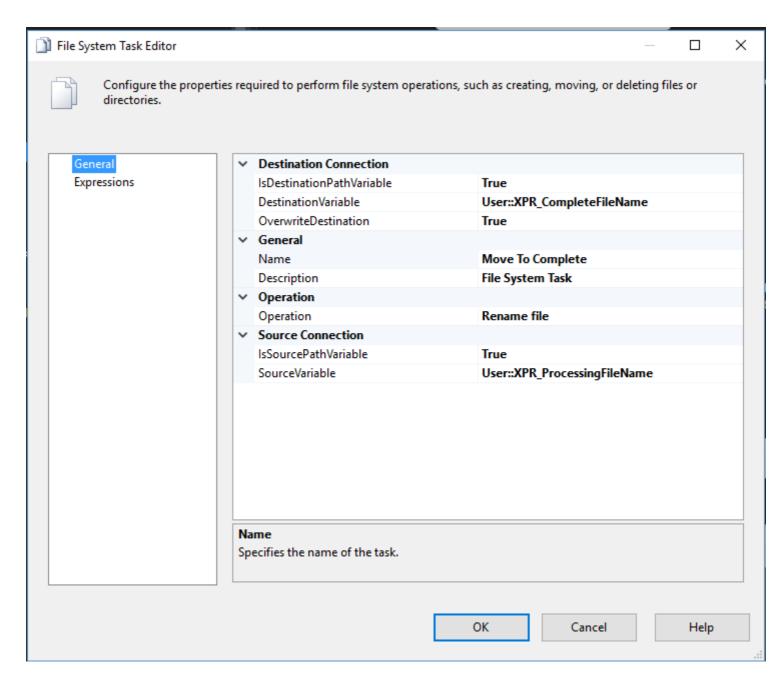
Once you've placed the task, double click to open it.



From here you'll want to first give the task a name. This helps later when you're reading logs looking for errors, you can recognize your file move task by name in those logs. In my case I named the task, Move To Complete.

Next you have two options to define your source and destination files. You can either define two file connections in your package. And then choose False for IsDestinationPathVariable and IsSourcePathVariable. You would then click the cell to the right of DestinationConnection and SourceConnection and choose the file connections from your package. I find more often I'm moving tasks in a loop, so I have a need to move more than one file per package execution.

If you want to be able to handle multiple files, change the IsDestinationPathVariable and IsSourcePathVariable to true. Then your File System Task Editor will change to look like the below image.



You'll need two variables defined in your package to hold the **full** destination file path (directory structure and filename) and **full** source file path. In my case I'm reading the variable XPR_ProcessingFileName for the source file and XPR_CompleteFileName for the destination file.

Finally, notice the Operation is "Rename file" rather than "Move file" since In my system, we append datestamps onto the end of the filenames to mark when they are processed by ETL. You could also change this option to Move file if you'd like, but renaming a file from one filepath to another is a move.

Read Move file from one folder to another online: https://riptutorial.com/ssis/topic/5888/move-file-from-one-folder-to-another

Chapter 8: Read from a CSV file

Introduction

Using SSIS to extract data from a CSV file and insert into a SQL Server table

Examples

Read from a CSV file and insert data into a table

First, you need to prepare the environment by creating the SQL Server table and the CSV file.

Run the script below in SQL Server to create the SQL table either on a new database or an existing one. For this example, I used my 'TrainingDB' database.

```
/* Creates table for Students.csv */
CREATE TABLE StudentDetails
(
    Surname varchar(50),
    Firstname varchar(50),
    DateofBirth datetime,
    PostCode varchar(50),
    PhoneNumber varchar(50),
    EmailAddress varchar(50)
)
```

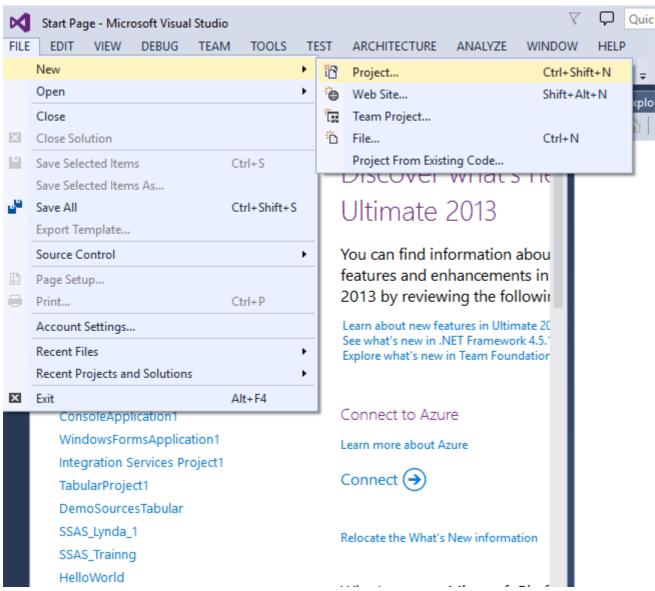
Now create a CSV file with the data below.

Surname	Firstname	DOB	Postcode	PhoneNo	EmailAddress
Bonga	Fred	24-02- 1990	SA1 5XR	08100900647	bonga.fd@gmail.com
Smith	Gill	08-05- 1992	RMT 12TY	08200900793	tastey0256@yahoo.co.uk
Taylor	Jane	01-12- 1979	PM2E 3NG	09600900061	jane0098@aol.com
Brown	John	06-10- 1986	CQ7 1JK	08200900063	johnbrown@hotmail.com
Cox	Sam	18-03- 1982	STR3 9KL	08100900349	cox.sam@gmail.com
Lewis	Mark	30-09- 1975	DN28 2UR	08000900200	m.lewis@mail.net

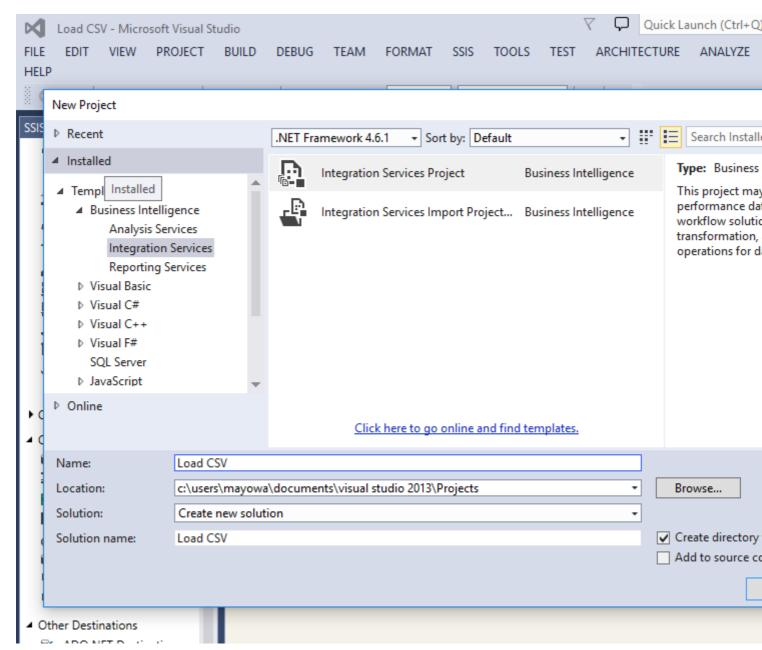
Surname	Firstname	DOB	Postcode	PhoneNo	EmailAddress
Kaur	Ahmed	26-07- 1984	NI12 8EJ	09500900090	kaur.ahmed@gmail.co.uk

You can copy this into Excel and save as a CSV file.

After launching Microsoft Visual Studio, navigate to File - New - Project, as shown below.

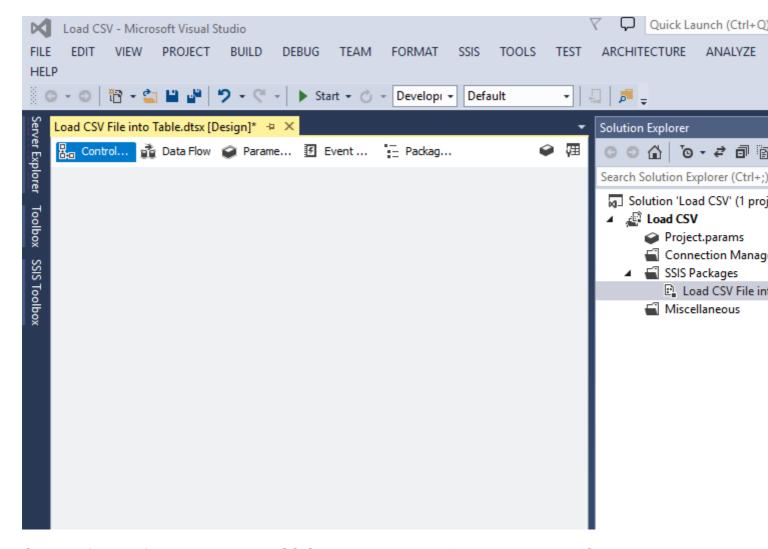


Under the Business Intelligence group, select Integration Services and Integration Services Project. Enter a name for project and a name for the solution, for example "Load CSV". You can check the "Create a directory for solution" box if you want to create a solution.



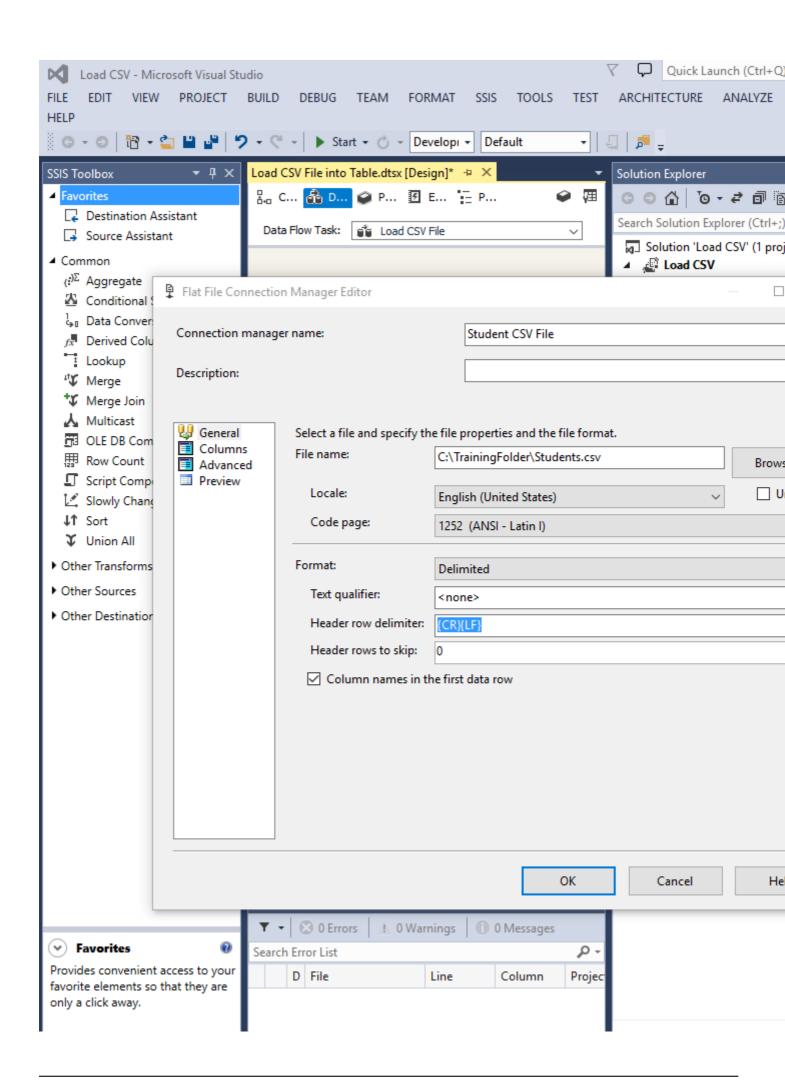
Click OK

On the right side of the displayed screen, in the "Solution Explorer" window, change the name of the default package to "Load CSV File into Table"



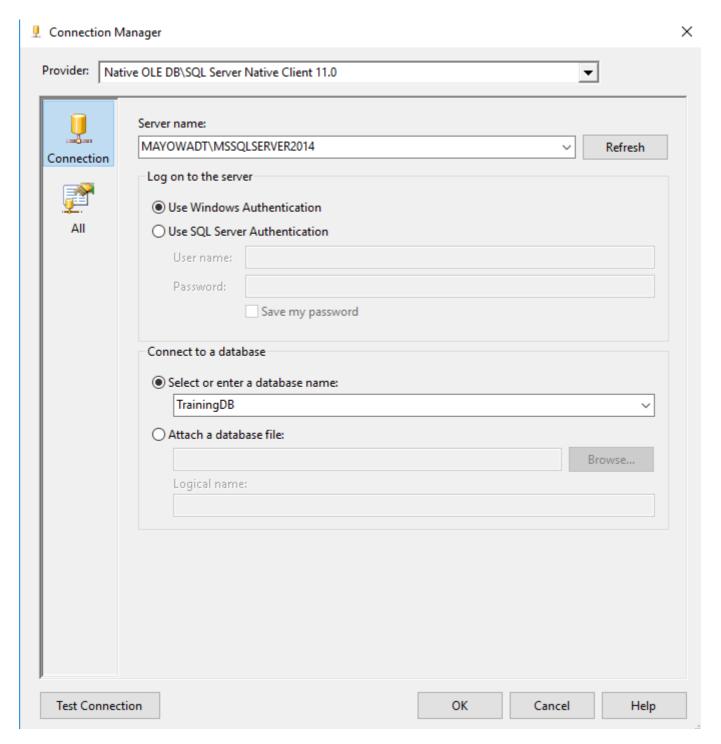
On the left side of the screen, in the SSIS Toolbar, drag the "Data Flow" to the "Control Flow" window and rename the task to "Load CSV File"

Next, you need to setup the connection managers for both the CSV file and the SQL Server table, which are also known as source and destination respectively. At the bottom of the screen, under Connection Managers, do a right click and select "New Flat File Connection" and configure the Flat file connection manager as shown below.



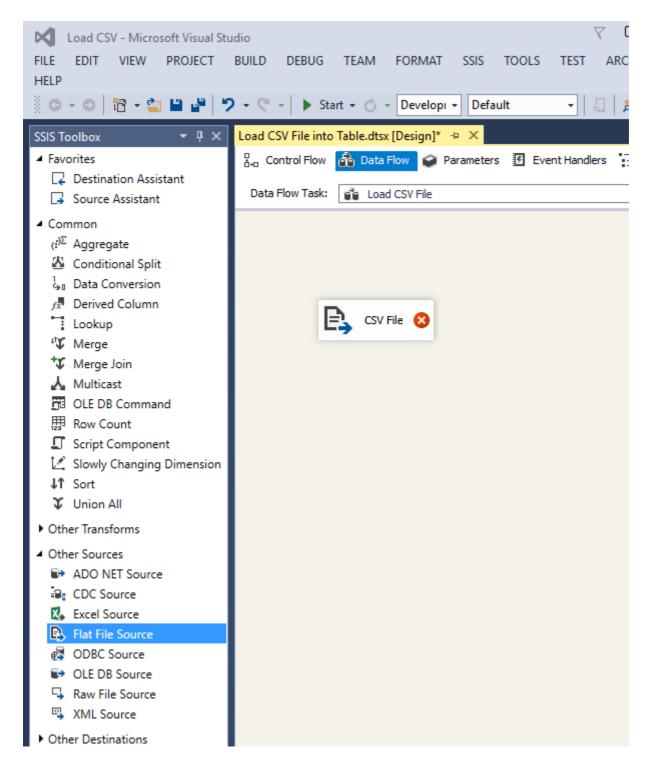
Enter a suitable Connection manager name and specify the filepath for the Students.csv file. Click OK.

For the table's connection manager, do a right click again in the Connection Managers window and click on "New OLE DB Connection". Click on New and specify the Server name and database name that contains the StudentsDetail table.



You can test the connection by clicking "Test Connection" then click OK and OK again. You should now have the 2 Connection Managers at the bottom of the screen.

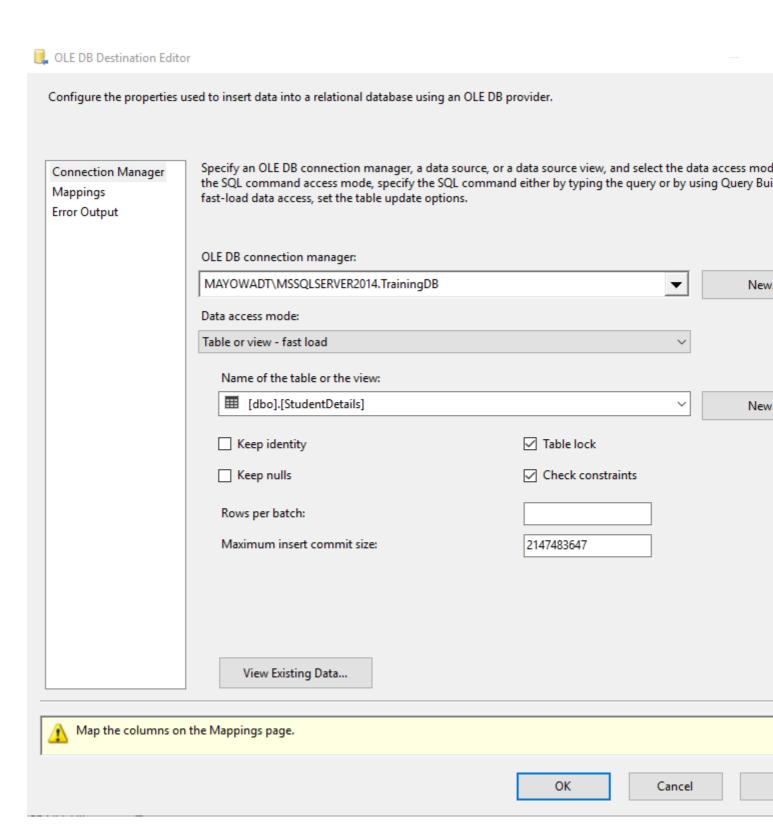
Drag the "Flat File Source" from the SSIS Toolbox into the "Data Flow" window and rename it as "CSV File".



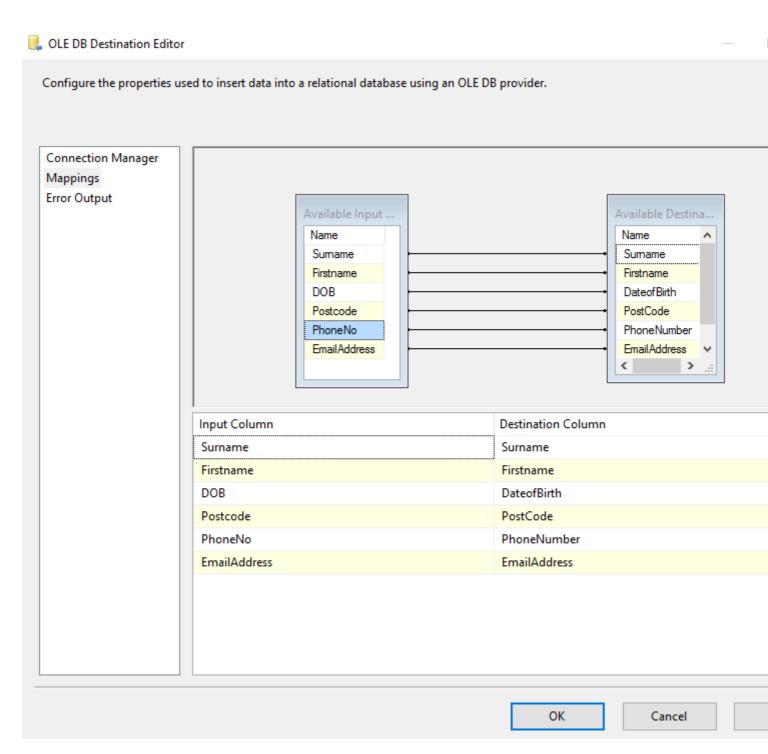
Double click on this source and select the "Student CSV File" connection manager. Click on Columns on the left side of the screen to review the columns in the file. Click OK.

Then drag the "OLE DB Destination" from the SSIS Toolbox to the "Data Flow" window and rename it as "SQL Table". Drag the blue arrow from the source to the destination.

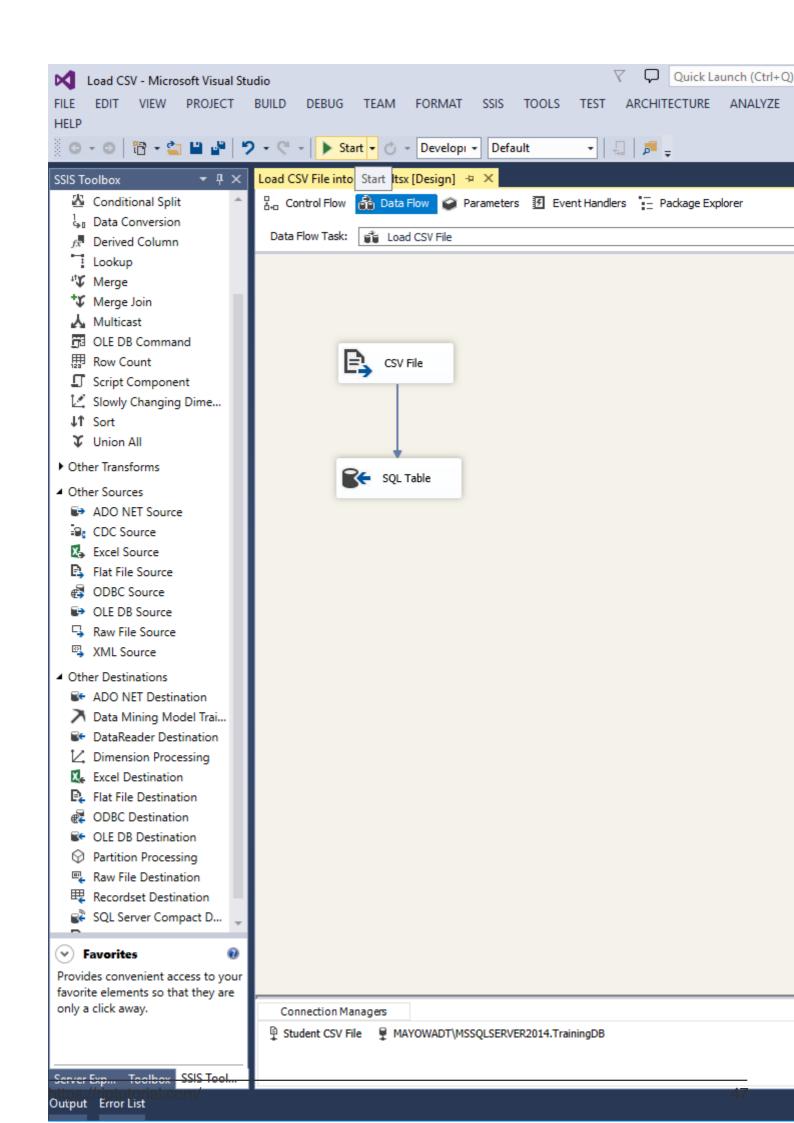
Double click on the destination and configure as shown below.



Click on Mappings on the left side of the screen and ensure all fields are mapped correctly from source to destination.



Click OK. Your screen should look like the image below.



Chapter 9: Sorting incoming data, but send forward only a subset of rows

Examples

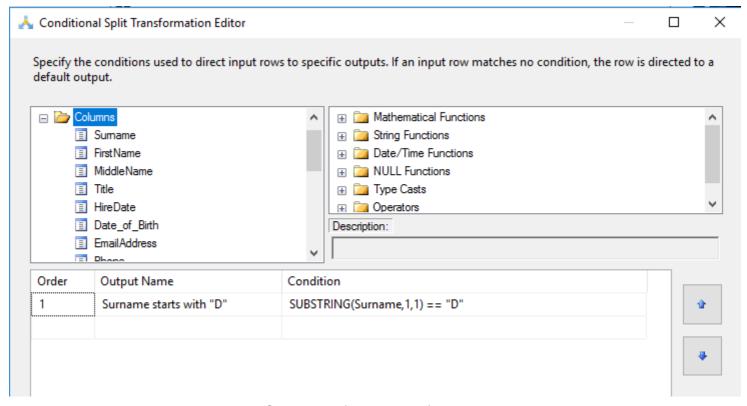
Using Sort and Conditional Split components

Since you need to sort & rename the fields, the best option will be the Sort Component in the Data Flow task (like you mentioned) . If you only want to rename columns, then use the "Derived Column" component. The Sort component should look as follows:

Sort Transformation Editor Specify the columns to sort, and set their sort type and their sort order. All nonselected columns are copied unchanged. Available Input Columns Name Pass Through **~** FirstName \checkmark LastName MiddleName \Box ~ \checkmark Title HireDate П ~ \checkmark BirthDate EmailAddress ~ Input Column Output Alias Sort Type Sort Order A Compariso LastName Surname ascending FirstName FirstName ascending 2 BirthDate 3 Date_of_Birth ascending

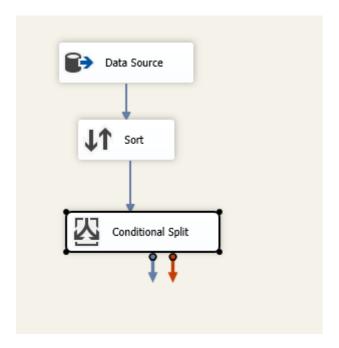
In my example, you can see the LastName, FirstName & BirthDate are sorted and LastName & BirthDate are renamed.

To return a subset of rows, you should use the Conditional Split component.



In my example, only rows that have Surnames (LastNames) that start with "D" would be returned.

The Data flow task should look like this:



Note: Sorting operation can be done in the database using SQL scripts (stored procedure) so it is advised to use SQL for better performance.

Read Sorting incoming data, but send forward only a subset of rows online: https://riptutorial.com/ssis/topic/9668/sorting-incoming-data--but-send-forward-only-a-subset-of-rows

Credits

S. No	Chapters	Contributors
1	Getting started with ssis	billinkc, Community, Rich
2	Check if a file exists	Ako, observer
3	Convert datatype from Integer in YYYYMMDD format to Date	Ako
4	Create a CSV file and write from SQL Server into that file	observer
5	How to use variables inside a script component	Hadi, observer
6	Load multiple CSV files of same format from a folder	observer
7	Move file from one folder to another	Shannon Lowder
8	Read from a CSV file	MayowaO
9	Sorting incoming data, but send forward only a subset of rows	MayowaO, observer