Microsoft Integration: SSIS Foreach Loop Container and It's Use

Aim: To copy any numbers of the file and particular file type from one directory to another directory.

In my case, SSIS junk file directory to SSIS junk file copy directory and the only text file is getting copied in another directory.

And all files name and location in stored in SSMS table.

Tools required:

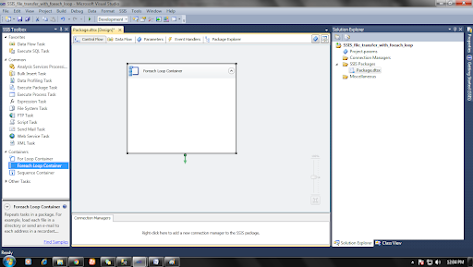
* SQL Server management studio.
* SQL Server Integration Service tool.
* A directory which contains your files (txt, CSV, Excel etc).
* A directory where you want to copy.

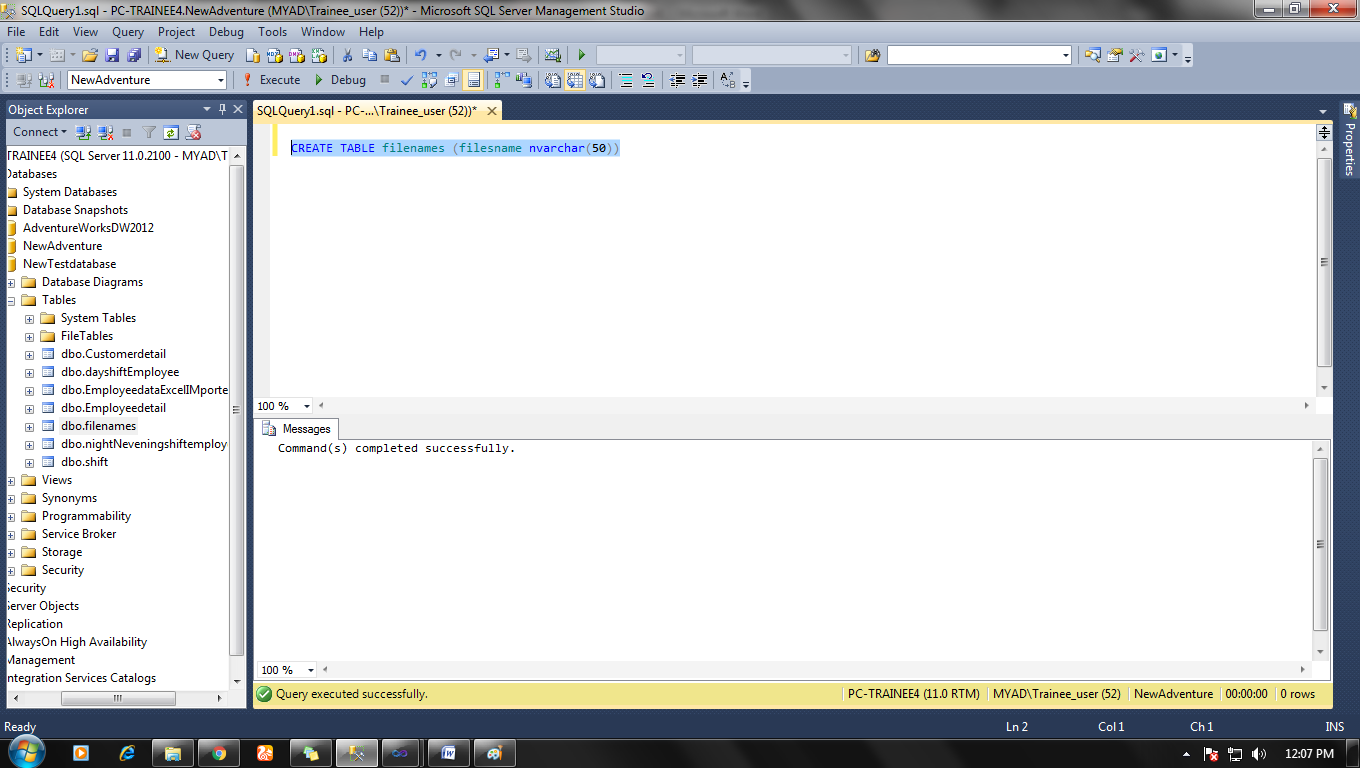
Steps to make a package which copy text file in another directory.

Step 1:

* Create an SSIS package with an appropriate name.

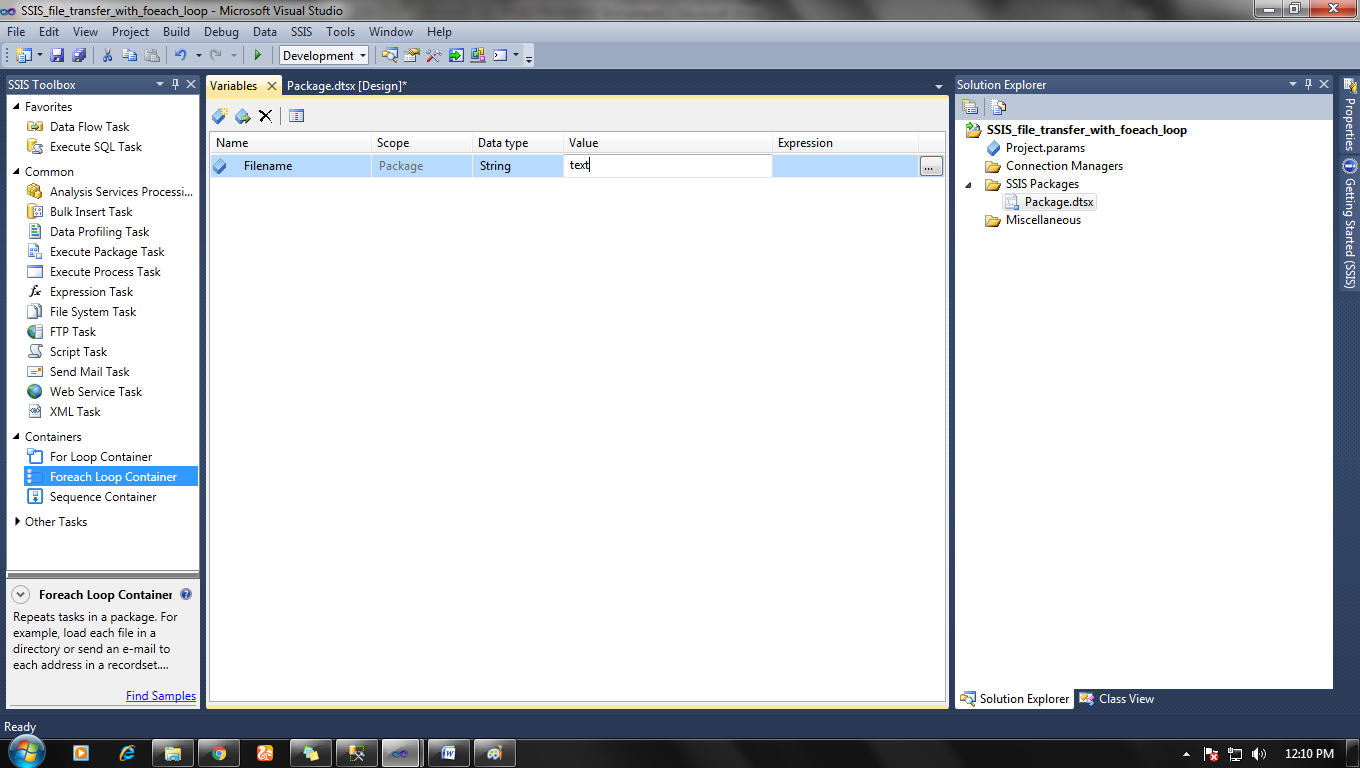
Set up your source file connection and destination SQL connection.

.

* Make an SSMS table to store file locations with name

**Step 2:**

Now in SSIS package create a variable name filename by clicking on a blue rubber sign icon in package designer on the upper right corner.

And a window will open then set up the variable type and initial value.

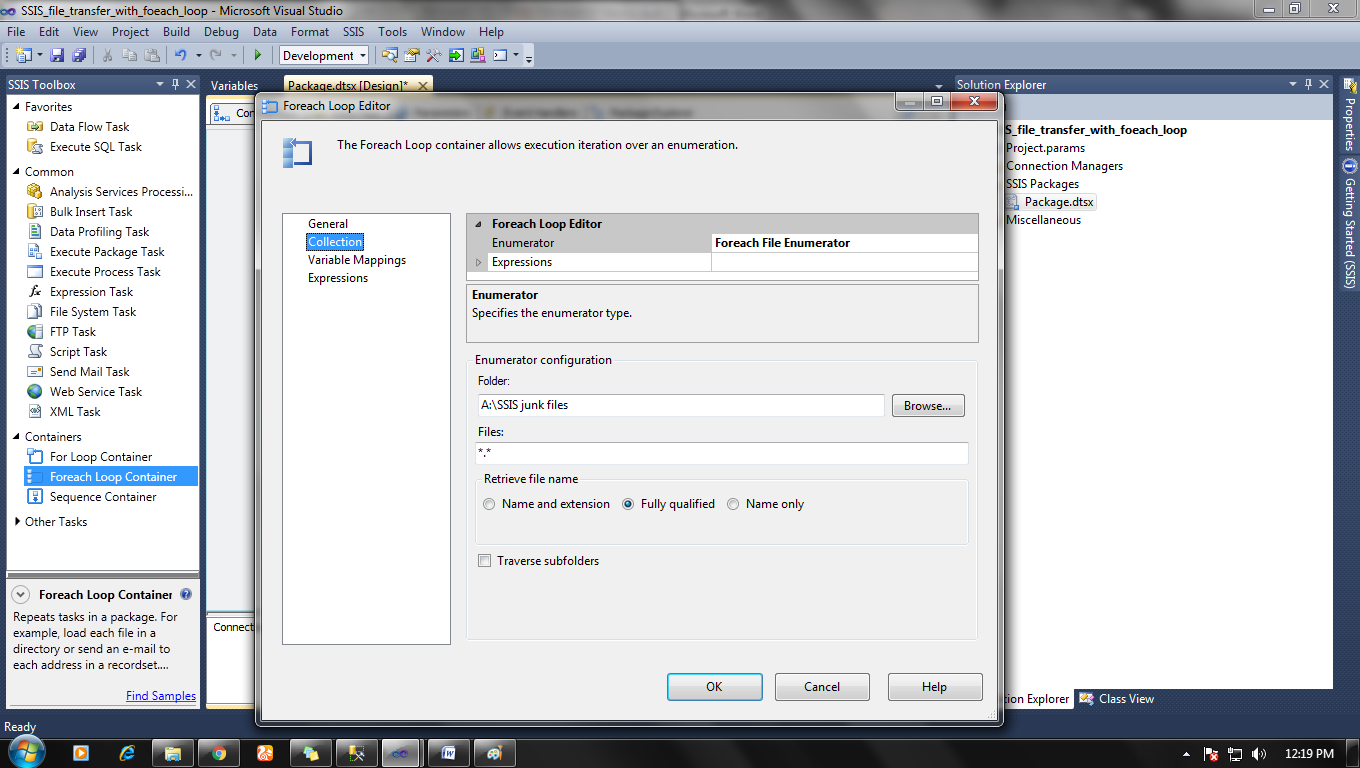
**Step 3:**

Now drag and drop foreach loop container in your control flow tab and double click on it.

A window will then go to collection option and set destination file path to you file directory and choose to retrieve file option as fully qualified.

Select Variable mapping>choose variable which you created  (filename variable) and select index option as 0.

And click ok.



**Step 4:**

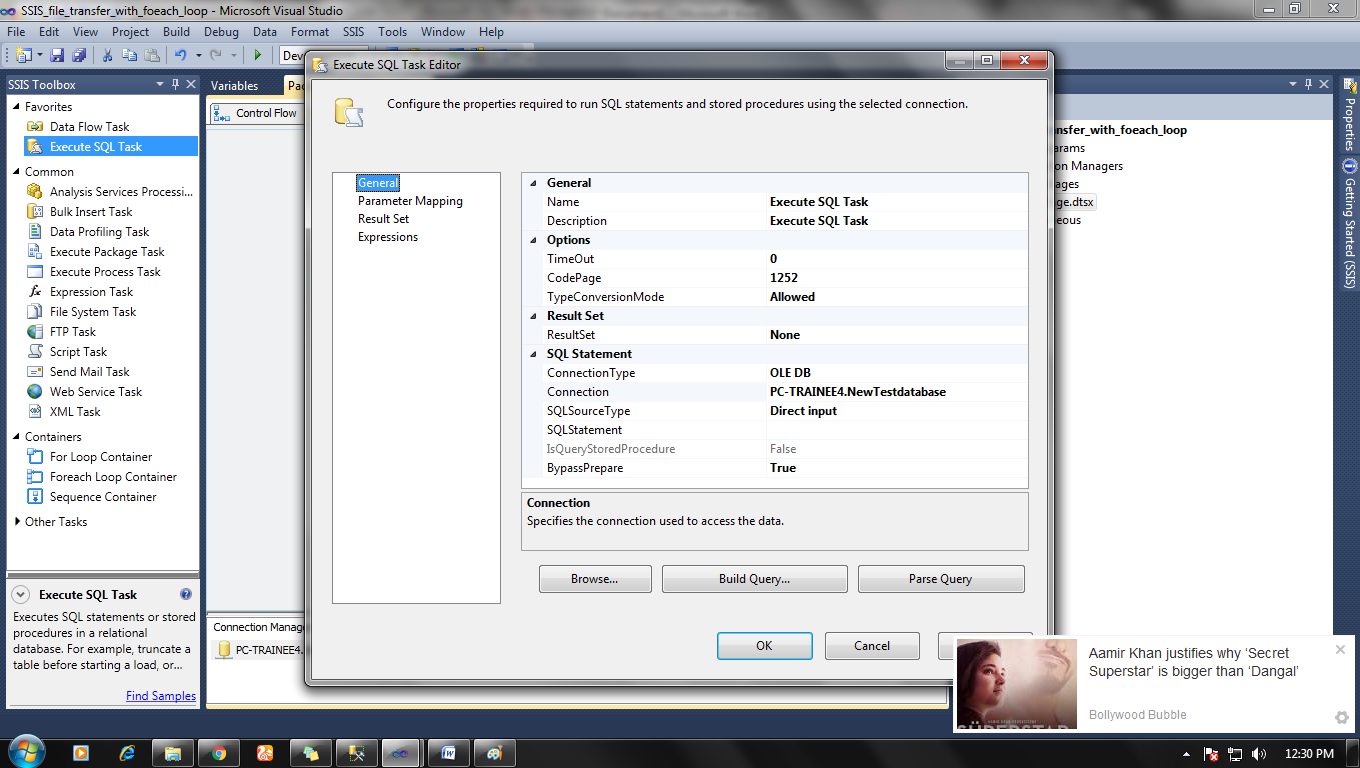
Now drag and drop Execute SQL task in Foreach loop container and double click on it.

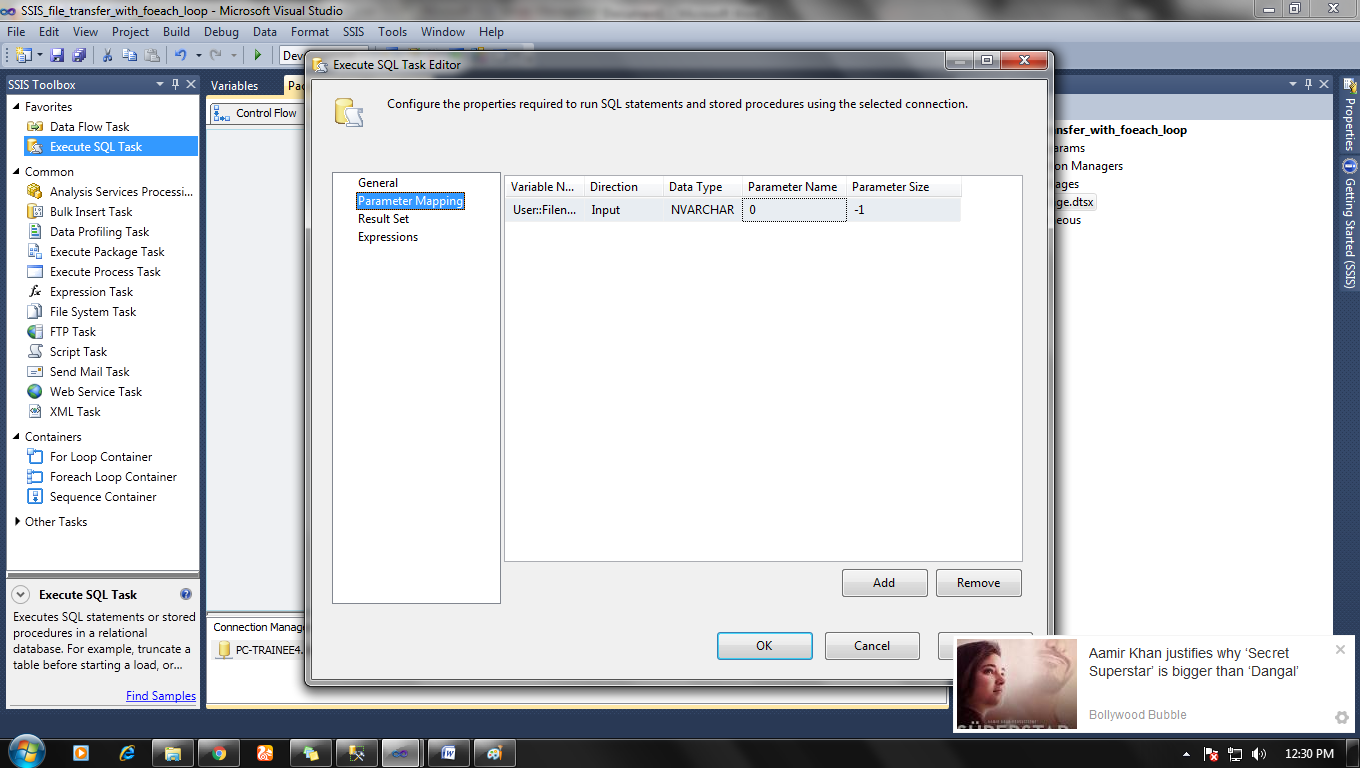
Set  General>connection> you OLE DB connection

And Set SQL statement as INSERT INTO Table name (column name) VALUES (?)

Now Parameter mapping >Variable name: Filename parameter> Direction: input> parameter name: 0> Parameter size: -1

Now click ok.





**Step 5:**

Select Common > File system task and drop it in foreach loop.

Double click on it.

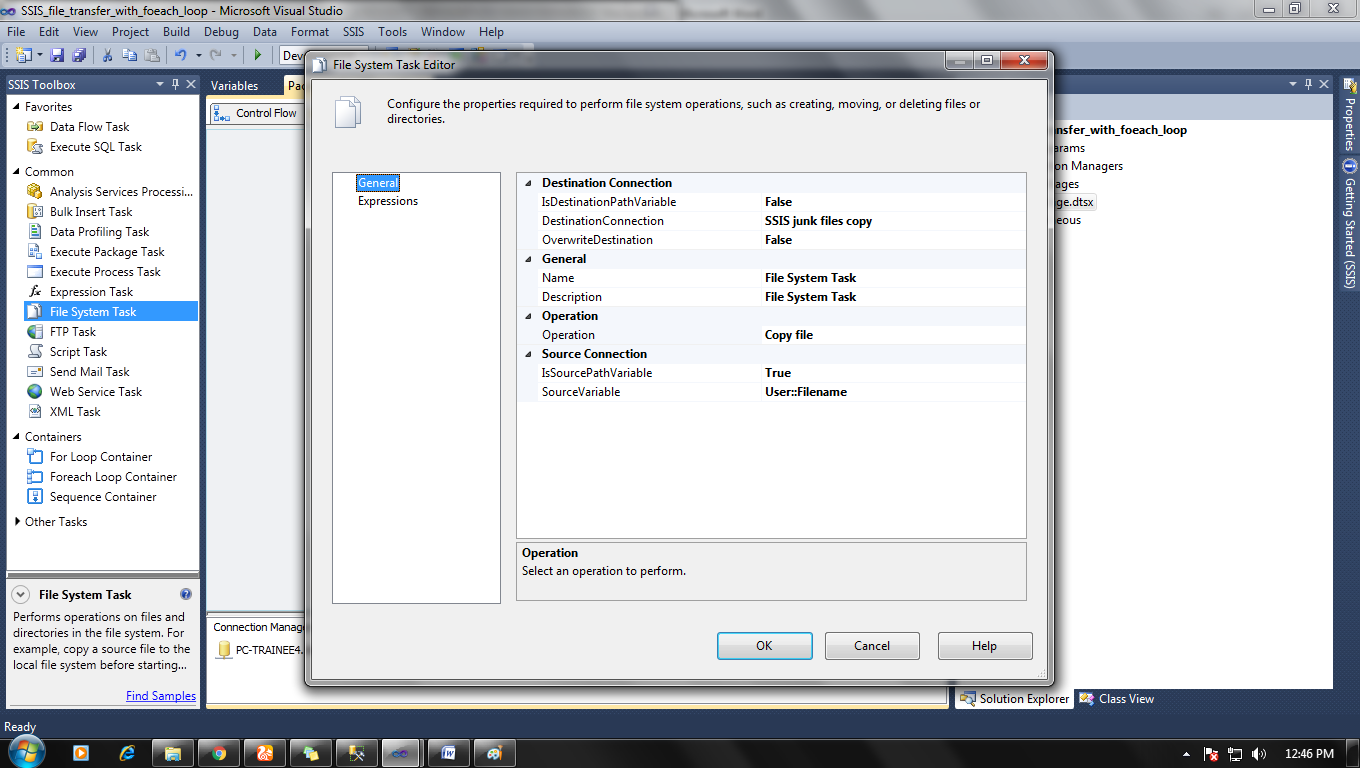
General> Destination connection: set up destination folder location

General> Operation: Copy file.

General> IsSourcePathVariable: True

General> Source variable: Your variable.

Then Click ok.



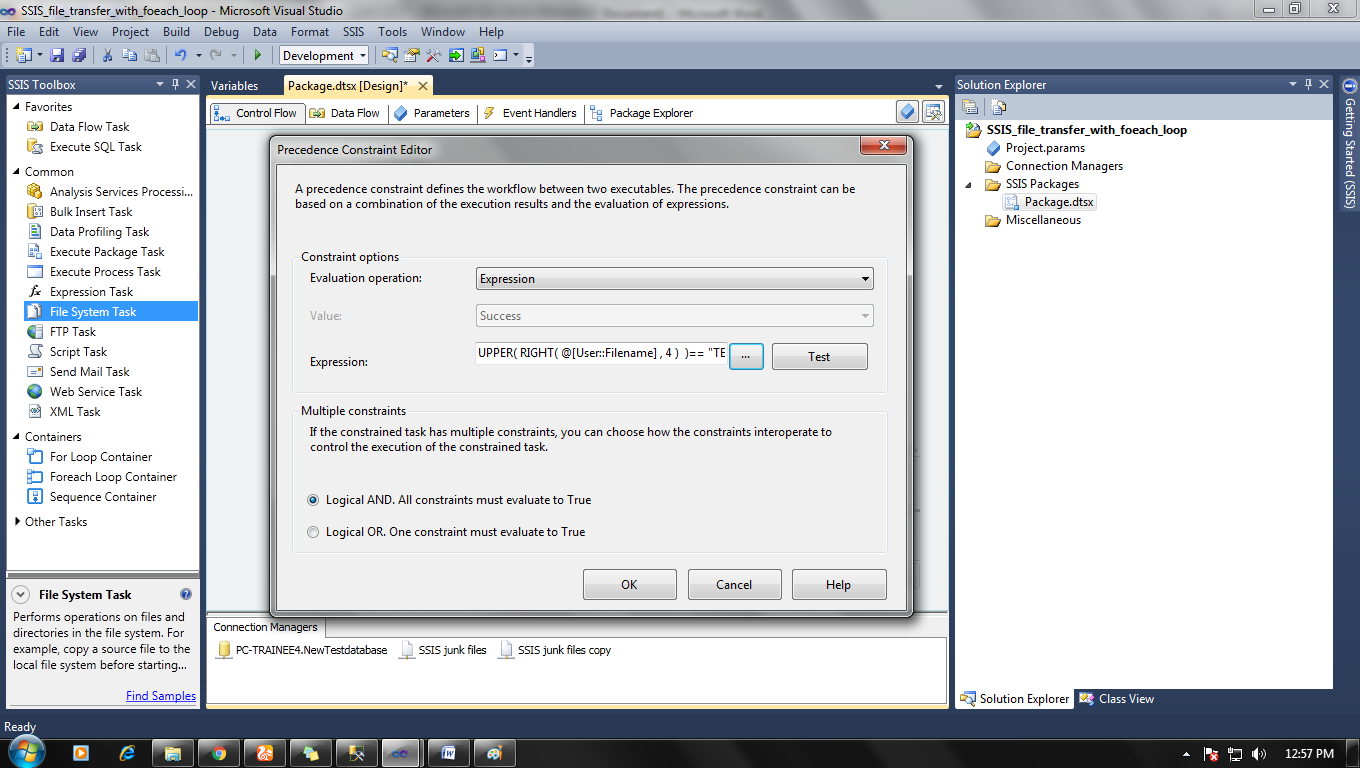
**Step: 6**

Now connect arrow connection between Execute SQL task and File system task.

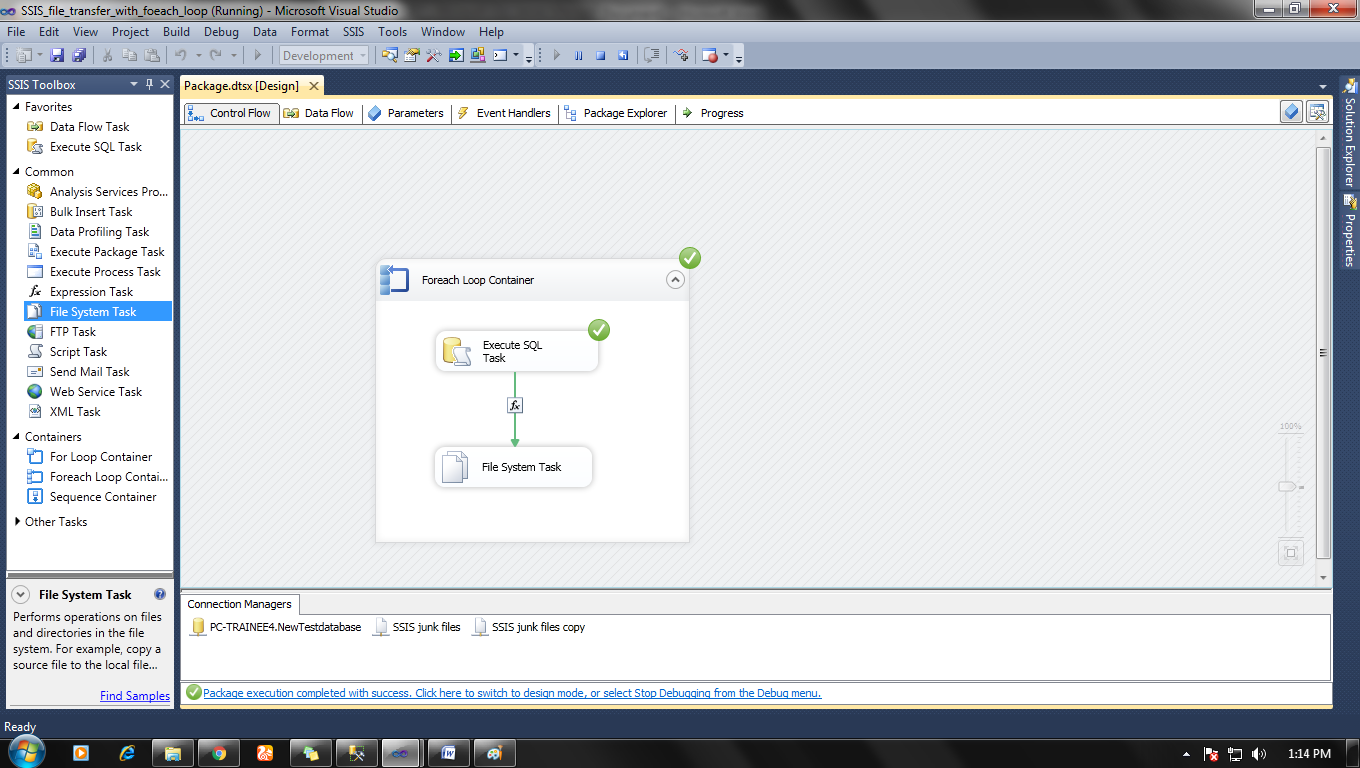
Right-click on arrow connection and choose edit and then

Evaluation operation> Expression

Expression: UPPER ( RIGHT( @[User::Filename] , 4 )  )== ".TXT"



Now everything is done so you can run your package and check your result.



Now if you run this task over and over again to load new file then some files would be repeated. So to overcome the problem of data overriding, you can add Execute SQL Task and File system task and Configure both to delete previous contain SQL table and Destination folder.