

Hands on Training Lab

SQLServer Integration Services

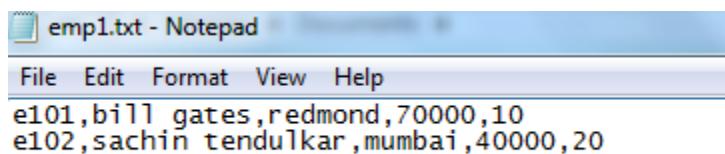
Table of Contents

LAB 1-DATA LOAD FROM FLAT FILE INTO RELATIONAL TABLE	3
LAB 2-DERIVED COLUMN TRANSFORMATION	8
LAB 3- AGGREGATE TRANSFORMATION.....	21
LAB 4- CONDITIONAL SPLIT TRANSFORMATION.....	28
LAB 5- LOOK UP TRANSFORMATION.....	34
LAB 6- COPY COLUMN TRANSFORMATION.....	48
LAB 7- MERGE TRANSFORMATION.....	54
LAB 8- MERGE JOIN TRANSFORMATION	60
LAB 9- UNION ALL TRANSFORMATION	63
LAB 10-SLOWLY CHANGING DIMENSION.....	65
LAB 11- EXTRACTING, TRANSFORMING AND LOADING DATA.....	73
LAB 12- EXECUTE SQL TASK & SCRIPT TASK	78
LAB 13-FILE SYSTEM TASK.....	86
LAB 14-FOR EACH LOOP CONTAINER WITH EXECUTE SQL TASK AND SCRIPT TASK	93
LAB 15- EXECUTE PROCESS TASK.....	100
LAB 16- USING ERROR REDIRECT ROW.....	107
LAB 17- TRANSACTION SUPPORT AND CHECKPOINT.....	110
LAB 18- USING DATA VIEWER.....	117
LAB 19- USING PACKAGE LOGGING.....	119
LAB20-PROJECT DEPLOYMENT	121
LAB21- PACKAGE EXECUTION USING SQL SERVER AGENT.....	130

Lab 1-Data load from Flat File into Relational Table

Objective	How to extract data from flat file and load into a relational table.
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Login details for connecting to the database

1. Create a separate folder “Project” for placing input data feed file
2. In that folder create a txt file emp1.txt and type this data

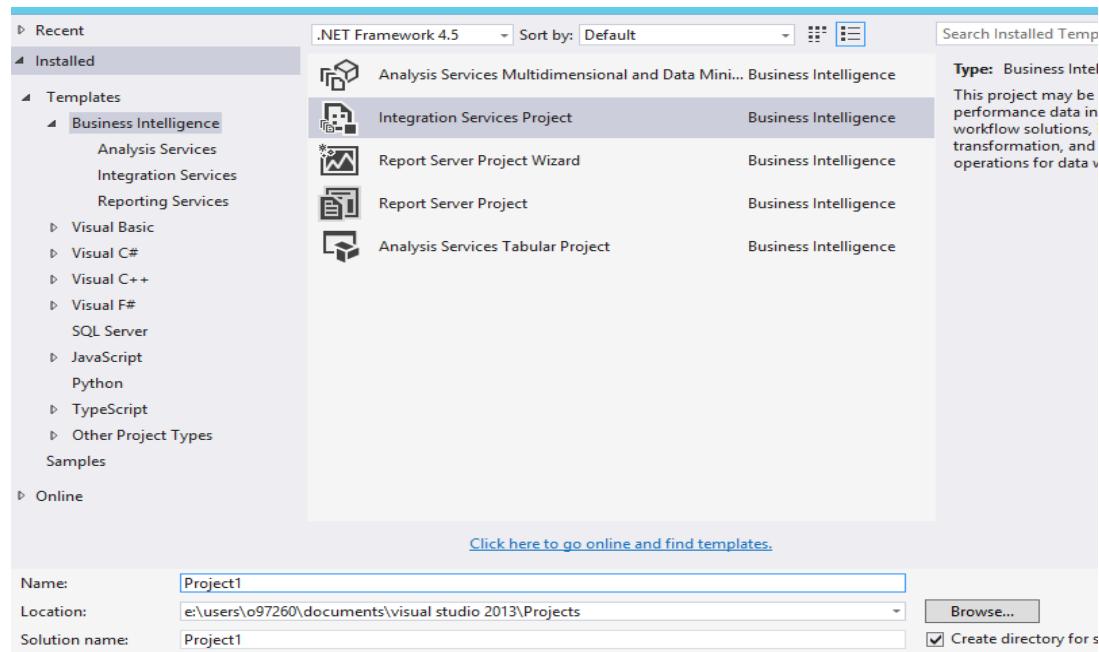


```
emp1.txt - Notepad
File Edit Format View Help
e101,bill gates,redmond,70000,10
e102,sachin tendulkar,mumbai,40000,20
```

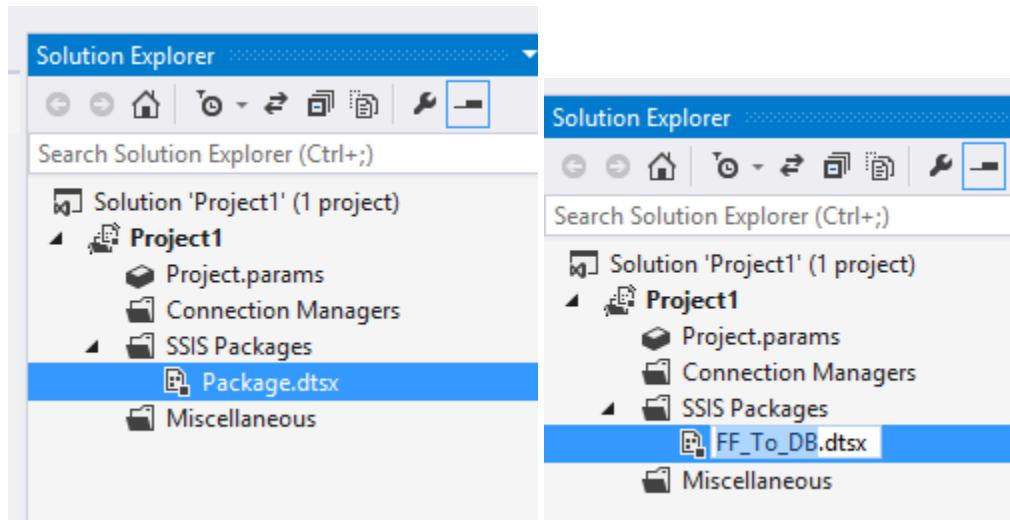
3. Connect to sql server using SSMS and create the following table

```
createtable emp1
(
    empno varchar(10)primarykey,
    empname varchar(50),
    city varchar(50),
    sal money,
    deptno int
)
```

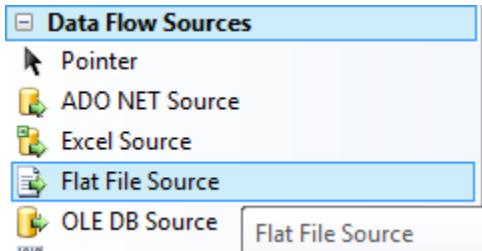
4. Start a new SSIS project. Start BIDS and click file => new project
 - o Select integration services project among Business Intelligence Projects.
 - o Rename it to “Project1”.



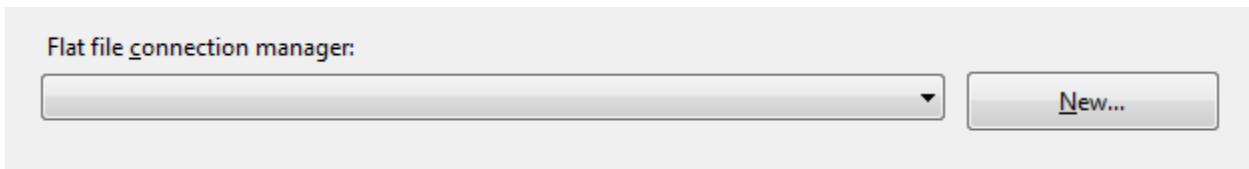
5. From Solution Explorer rename package.dtsx to FF_To_DB.dtsx



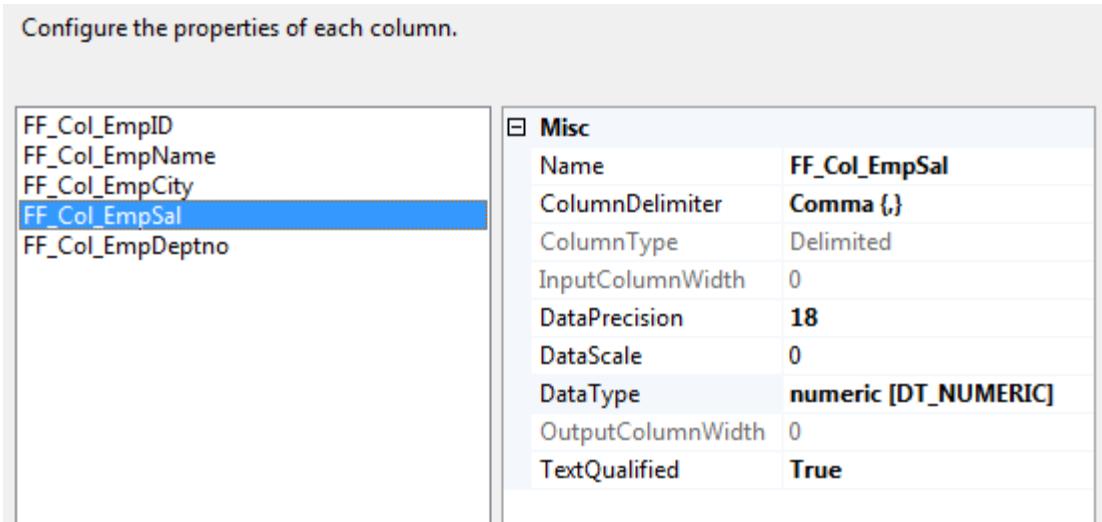
6. In the package designer, goto data flow tab and click the link in the middle to create a new data flow task. (this data flow task is automatically called from control flow).
7. In the data flow task designer drag “Flat File Source task” from toolbox
 - o Rename it “Emp Flat File Data”



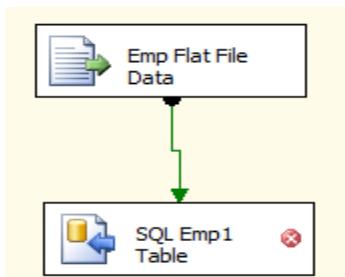
8. Double click the above dragged task to popup a dialog box. Click “New” Button to create a FF connection Manager.



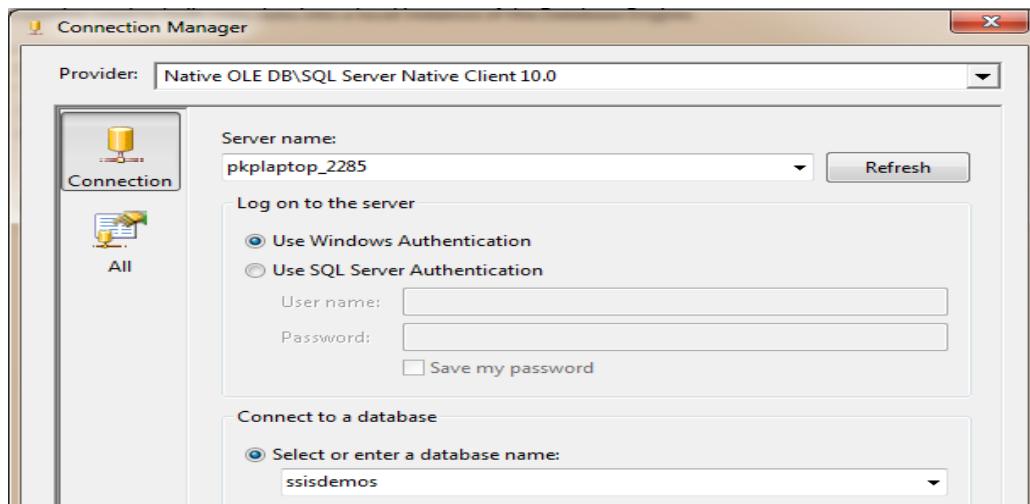
9. Give Connection Manager Name as: FF_Emp_CM
10. For filename, browse and locate emp1.txt.
11. Click Columns TAB to ensure columns identified properly
12. Click Advanced TAB and rename cols as below:
13. For Sal field set data type “currency [DT_CY]”
14. For Deptno field set data type “four-byte signed integer [DT_I4]”



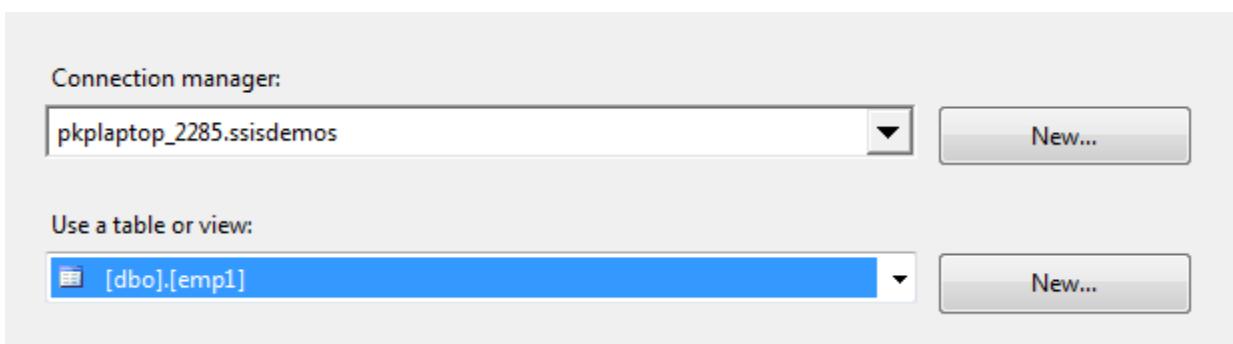
15. Click OK
16. In Toolbox , from Destinations section drag sql server destination
o Rename it as SQL Emp1 Table
17. Connect the green connector of Flat File Source to SQL Emp1 Table destination.



18. Double Click SQL Emp1 Table destination task.
- o Click New to create New Connection Manager, again click new
 - o Give the connection details as appropriate



19. Test Connection, Click Ok twice.
 20. Select the emp1 table created earlier in the dialog box as below:



21. Go to Mappings tab and map columns as below:

Input Column	Destination Column
FF_Col_EmpID	empno
FF_Col_EmpName	empname
FF_Col_EmpCity	city
FF_Col_EmpSal	sal
FF_Col_EmpDeptno	deptno

22. Click Ok to come to data flow designer surface
23. Click Run Button  on toolbar, the package should run successful. Click Stop Button explicitly on toolbar.
24. Goto SSMS and check the sql server table data using [select*from emp1](#)

Lab 2-Derived Column transformation

Objective	To load the data from excel sheet to OLE DB database.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS Project and connection to database

Source:

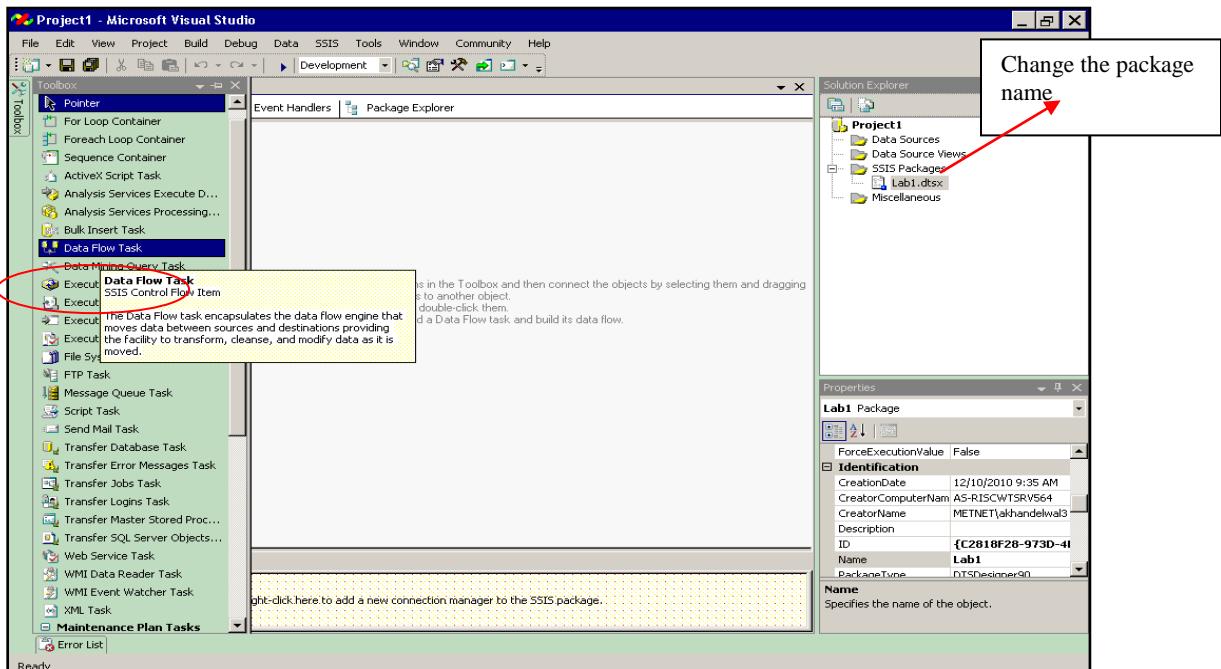
Microsoft Excel - Employee details					
	A	B	C	D	E
1	Emp code	Emp Name	Address	Age	Salary
2	101	a	xyz	23	20000
3	102	b	abc	43	56000
4	103	c	pqr	34	40000
5	104	d	asd	56	59000
6	105	e	abc	33	39000
7	106	f	def	27	29000
8	107	g	ghi	47	49000
9	108	h	jkl	55	70000
10	109	i	mno	28	30000
11	110	j	lop	38	36000
12	111	k	stu	49	50000
13	112	l	wwx	29	31000
14	113	m	yz	30	34698
15	114	n	qwe	40	49875
16	115	o	rtyu	50	58000
17	116	p	fnfd	39	39785
18	117	q	sfh	25	26000
19	118	r	jkg	26	27154
20	119	s	mnb	36	36987
21	120	t	afgh	47	46987
22					
23					
24					
25					
26					
27					
28					
29					
30					



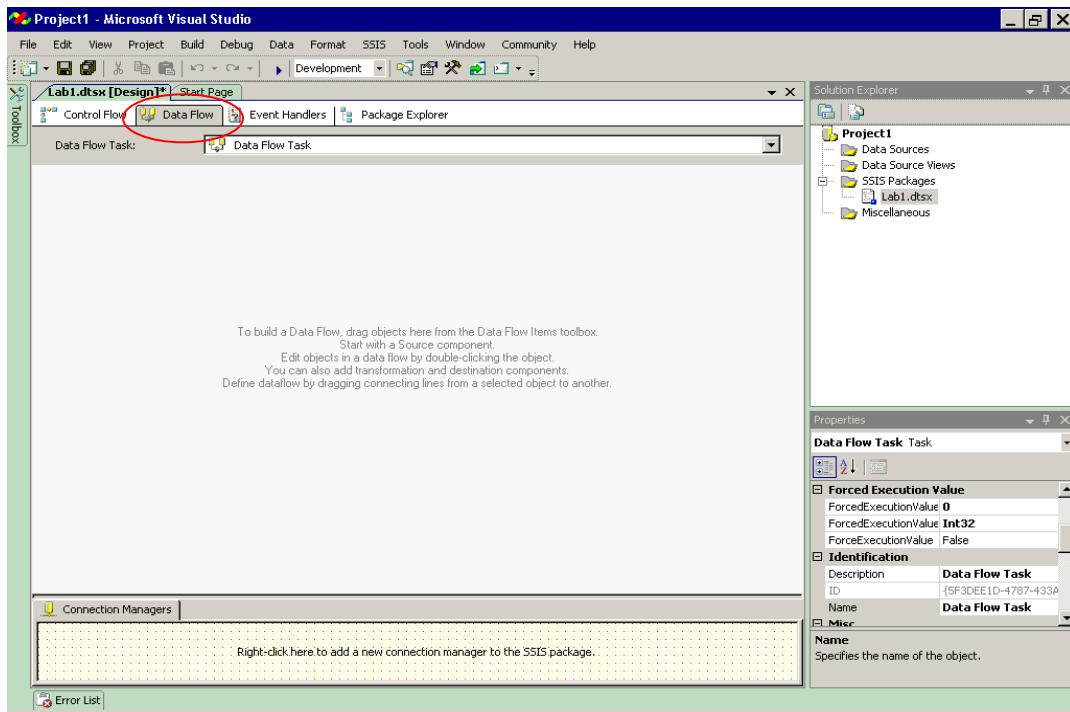
Employee_data.csv

Source File attached here:

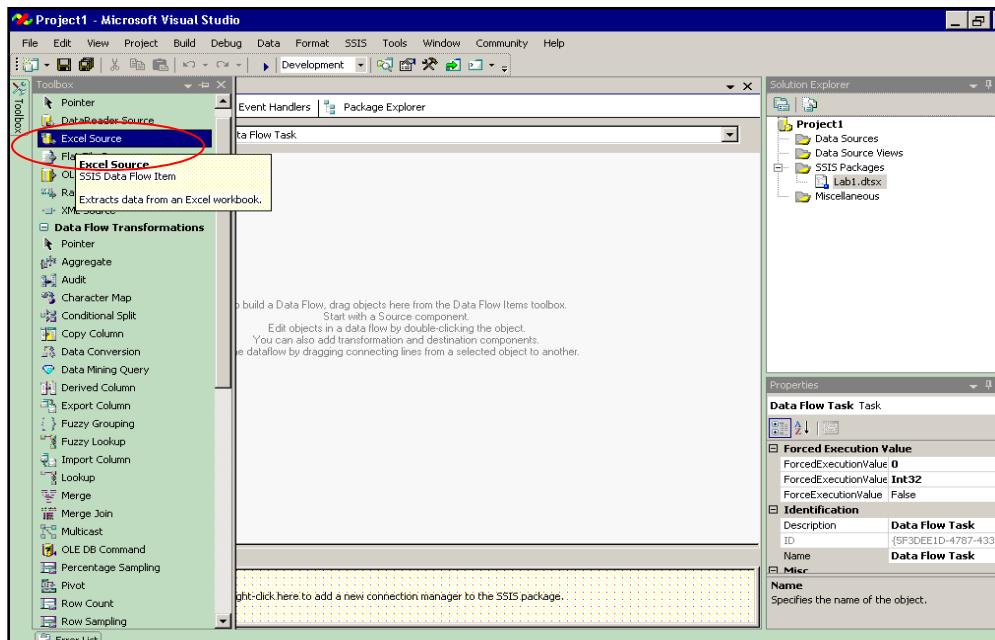
- 1) Create a new package named "Lab1" under the already created project.
- 2) Click on the toolbox and drag and drop the Data flow task on control flow task:



- 3) Double click the data flow task or click on the data flow task and select the data flow from the tab to begin the mapping:

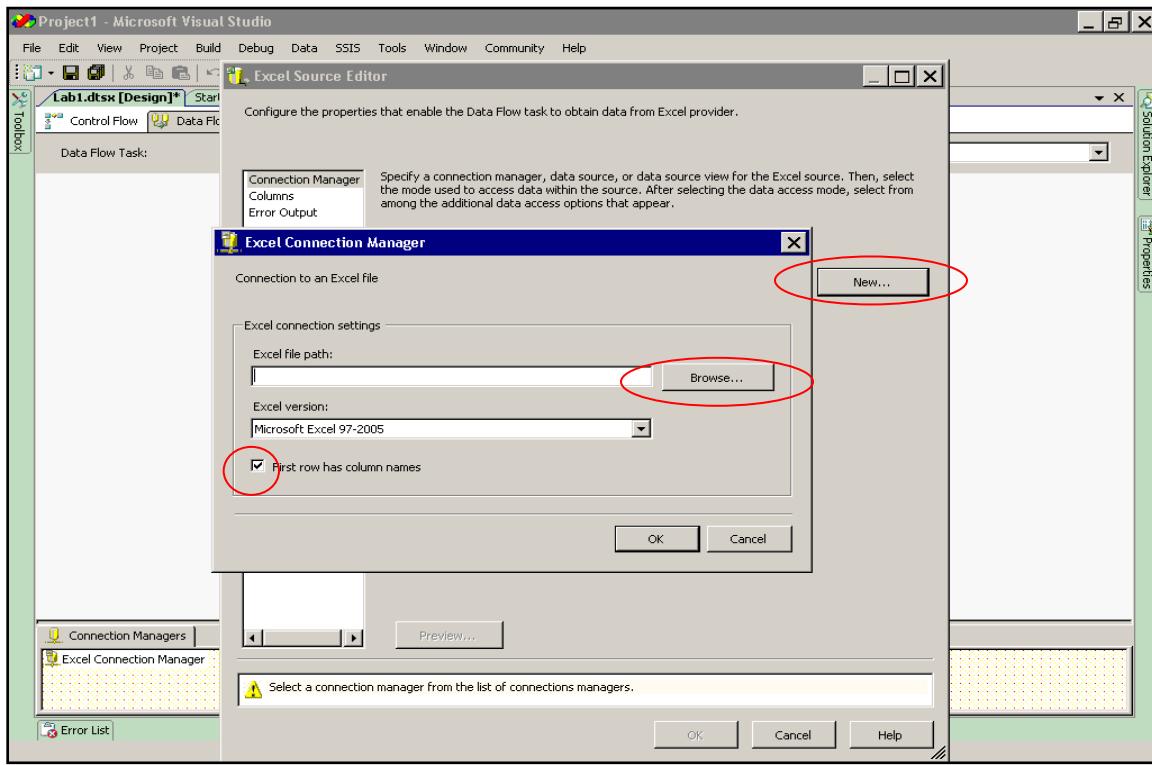


4) Select a source from the toolbox. Here we are taking the excel source:



5) Double click the source to open the editor:

5.1) Create a new connection manager:

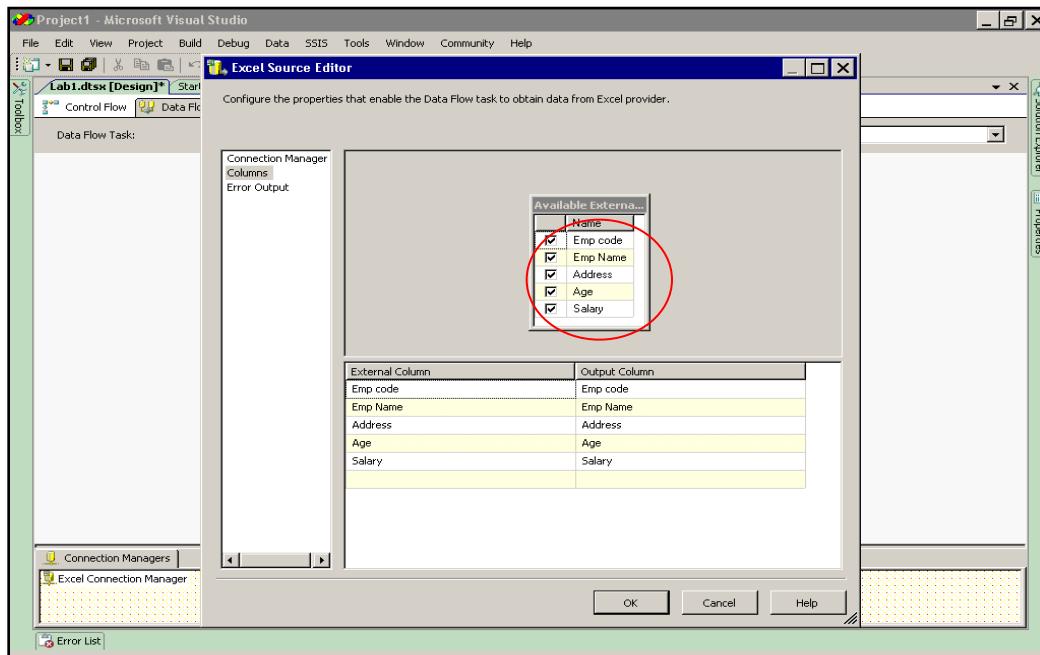


If the excel file contains columns then you must check the box for “First row has column names” check box.

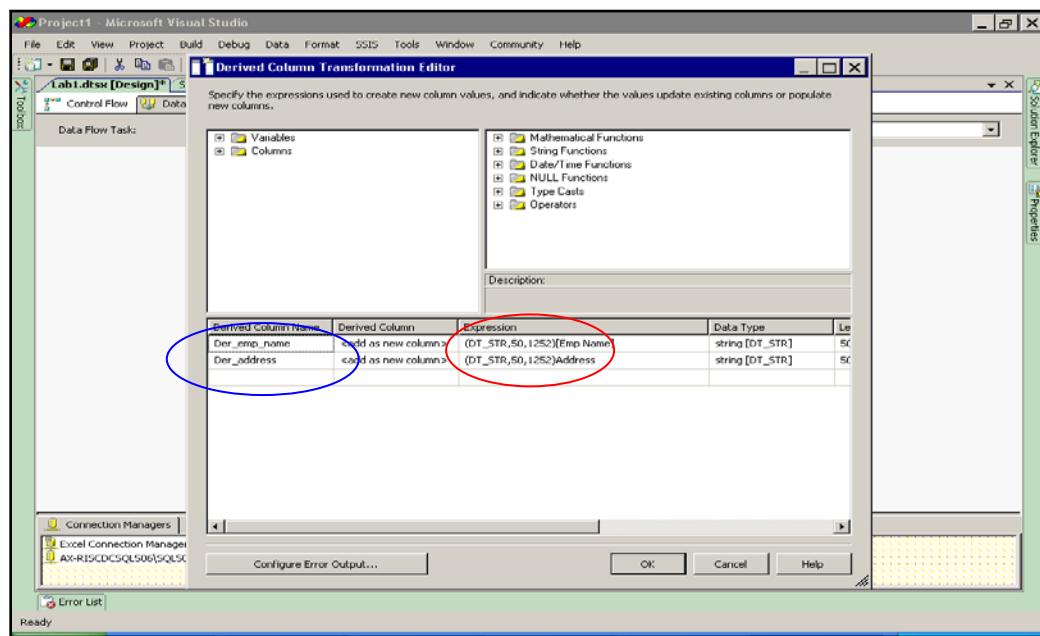
Click ‘OK’ after selecting the file and select Data access mode as – Table or view and name of the excel sheet as ‘sheet1’ if the data is in that sheet.

You can also click the “preview” button to view the data in the source table.

5.2) Click on columns to select the required columns in the target table:

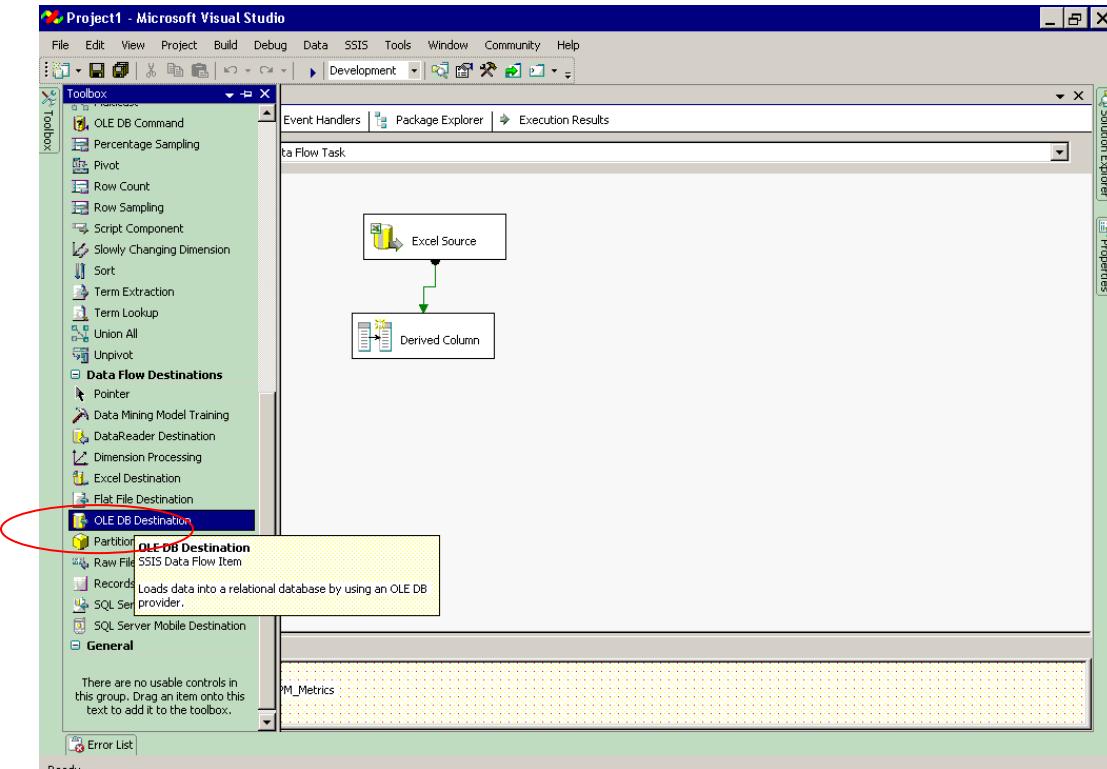


6) Select Derived column transformation from the toolbox and double click it to get the editor:



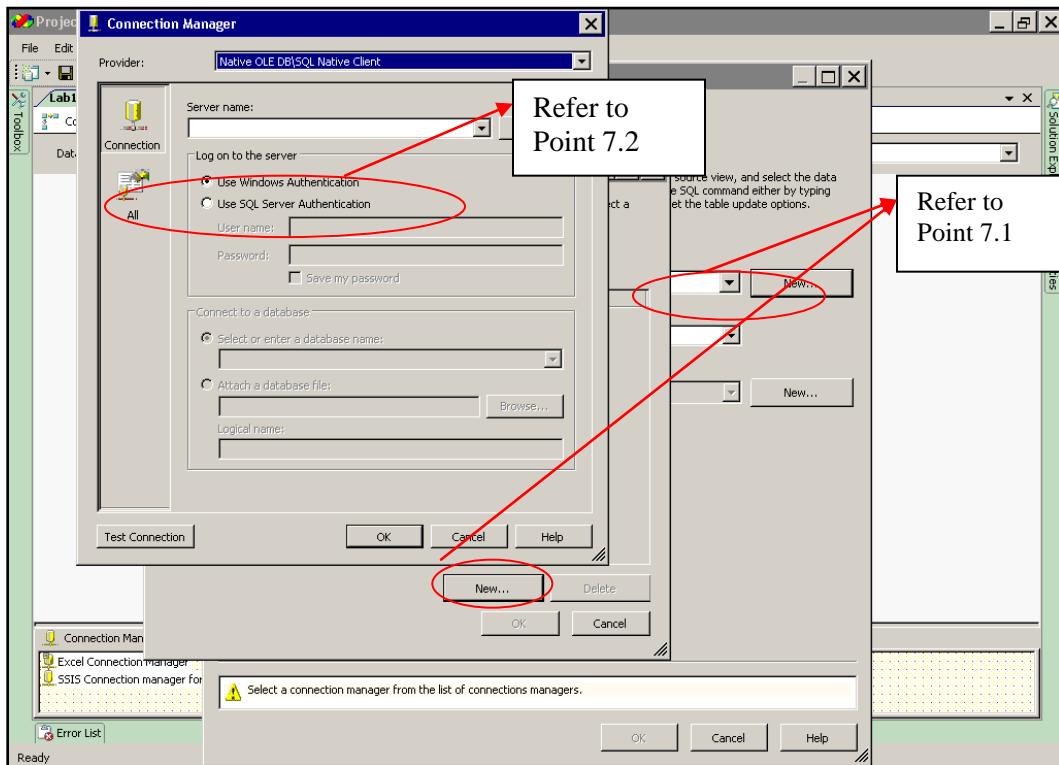
Change the data type of Emp_name and address (Unicode in source) to string and click OK.

7) Create a target table in the database (SQL Server) if not present, to transform the data:
Select OLE BD Destination as target table:



Connect Derived column transformation and target using the GREEN connector:

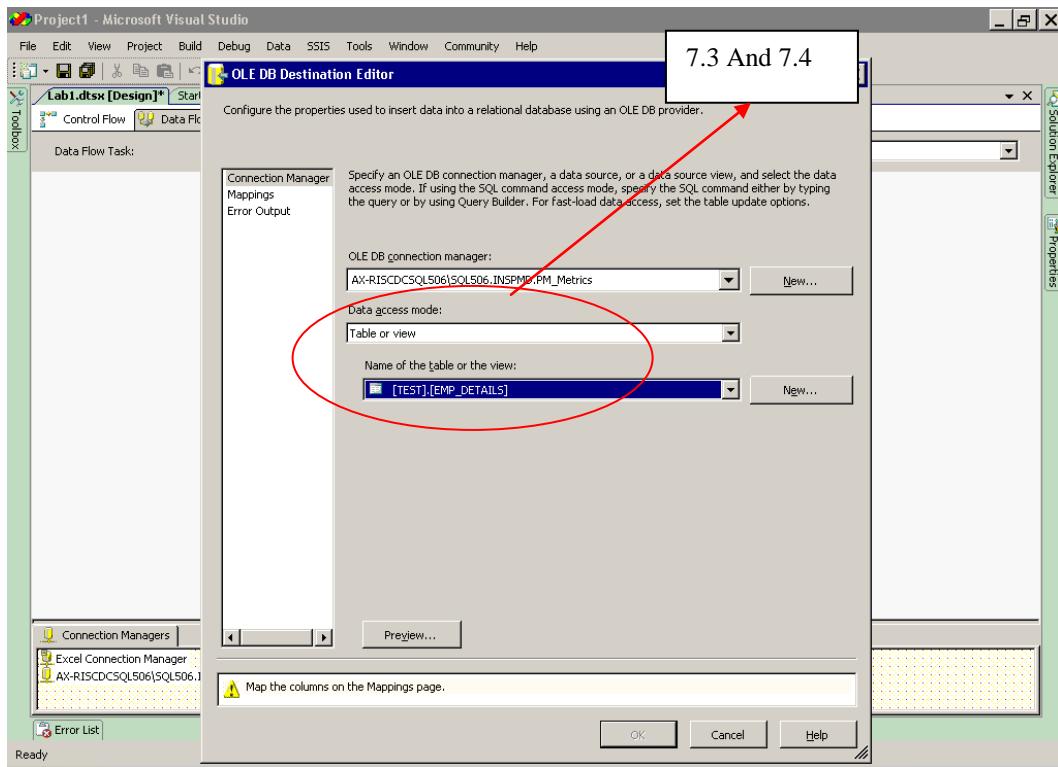
- 7.1) Double click OLE DB destination to open the editor. Create a New Connection Manager:



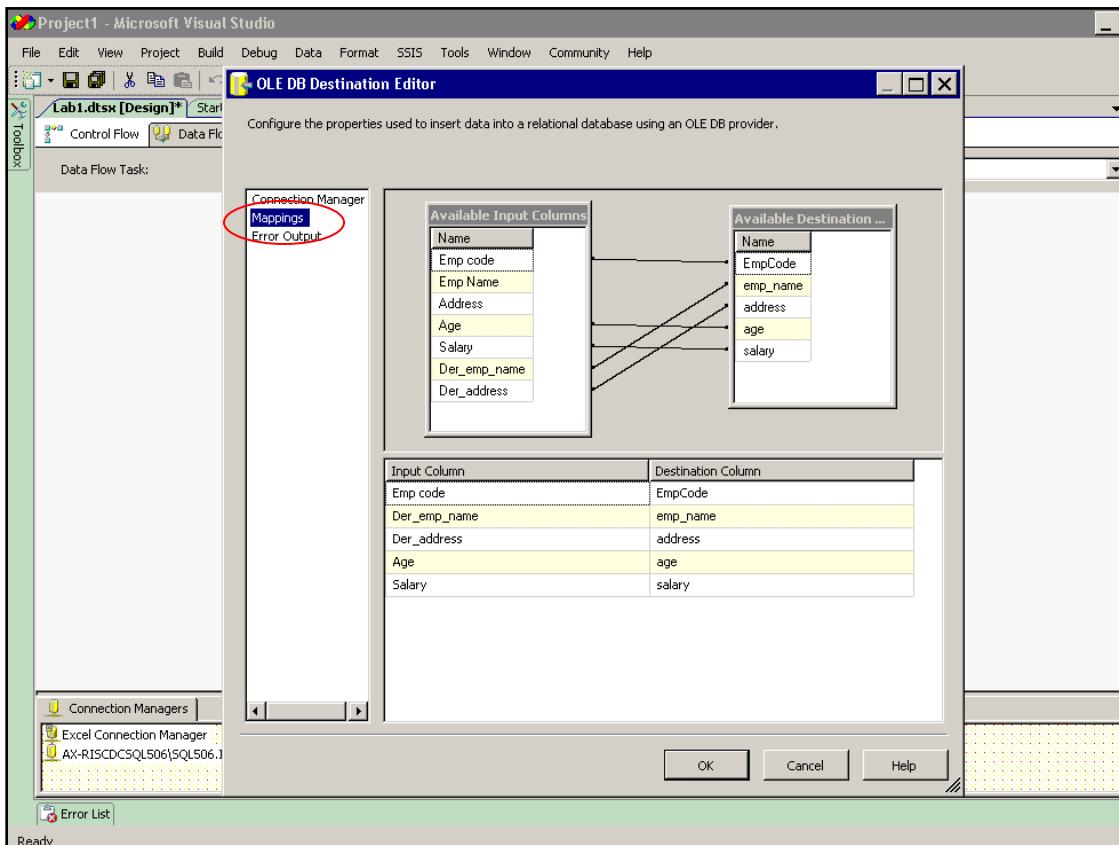
7.2) Select the server name, use SQL Server Authentication (use valid user name and password) and select a database name and test the connection.

Click OK till you are able to view the editor with a connection manager.
7.3) Select Data Access Mode as- Table or view.

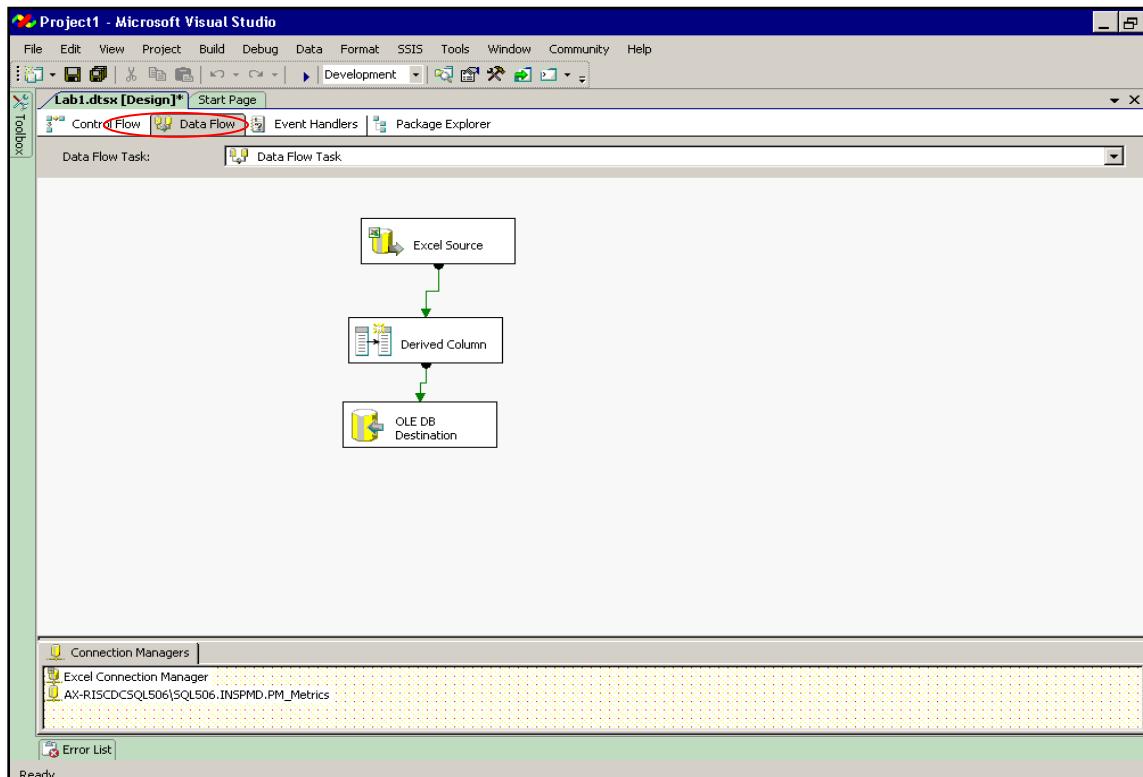
7.4) And then select the name of the target table.



7.5) Click on Mappings to map the data from source to target.

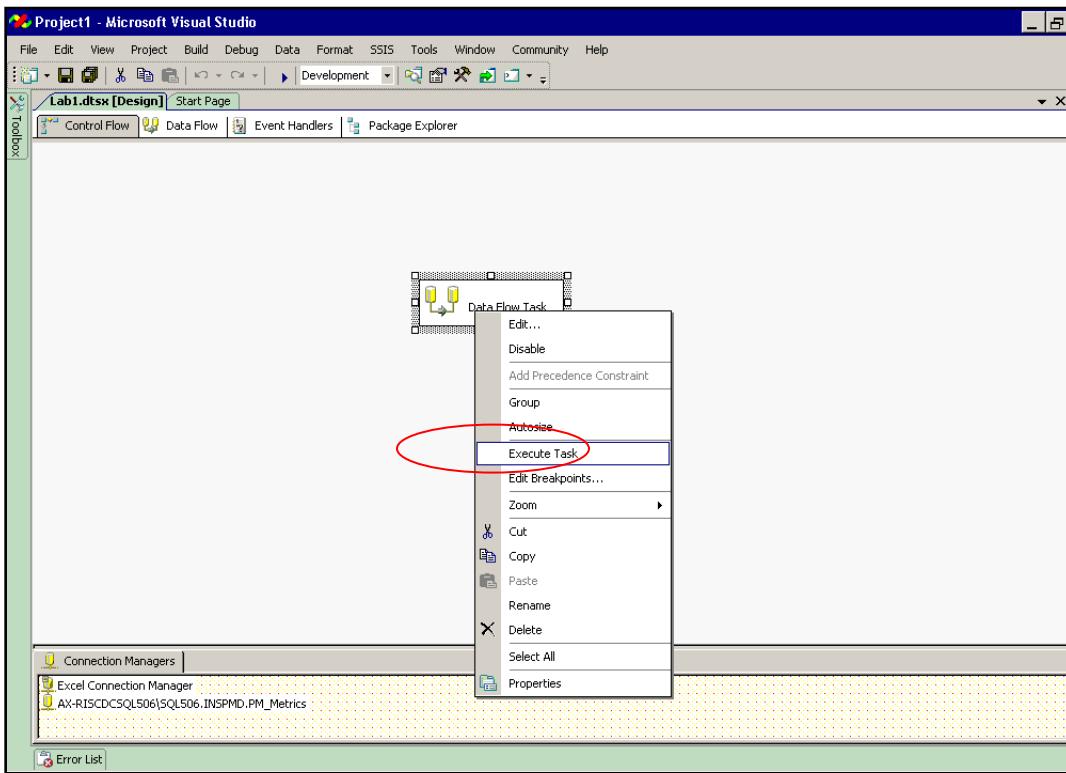


The mapping from source to target would be like:

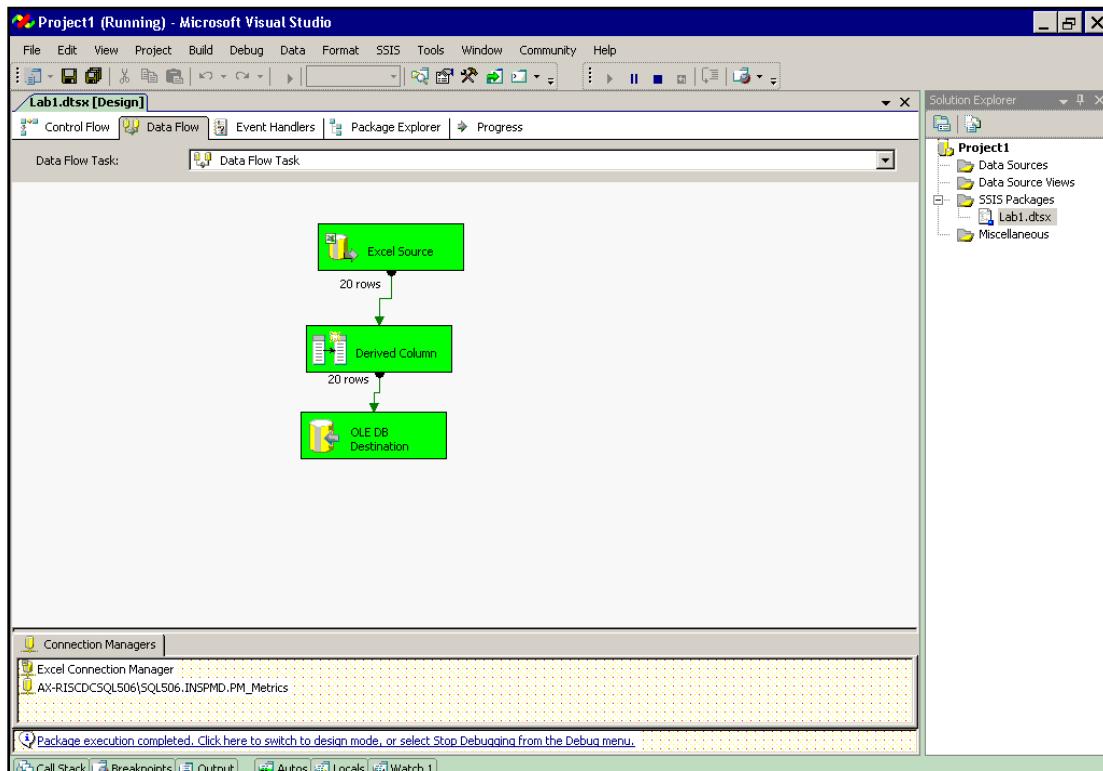


8) Save the project.

Switch to the control flow flow task and right click on data flow task. Click on Execute task.



9) After executing the data flow and on successful completion the Data Flow will be as follows:



You can check the data in the target table using SQL Server.

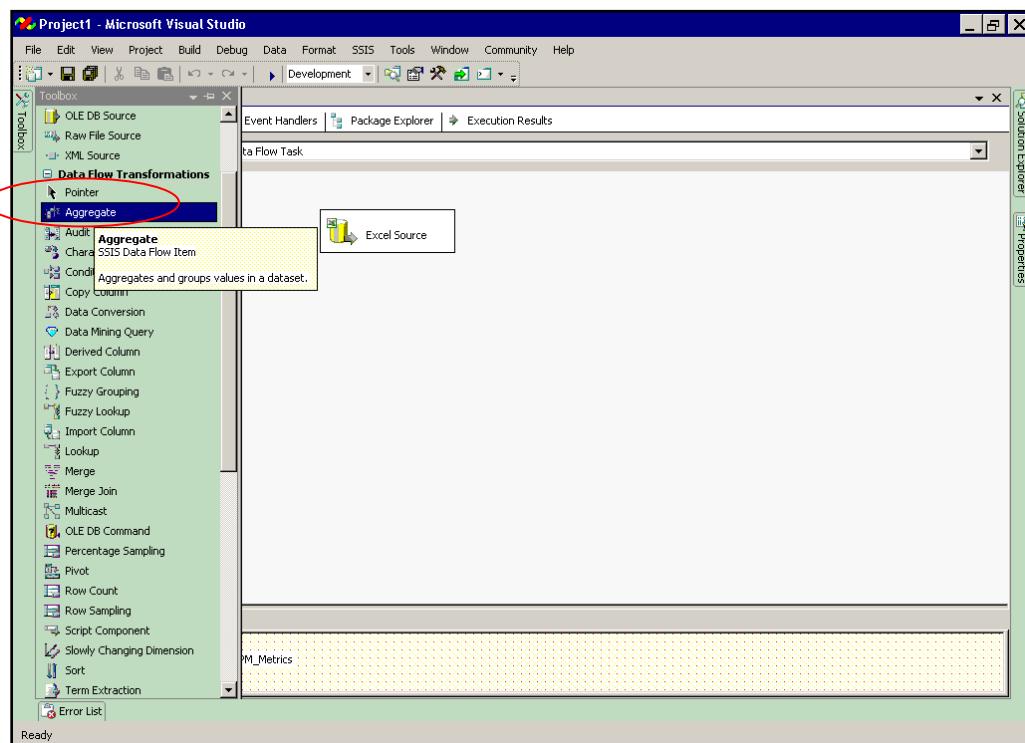
Emp_code	emp_name	Address	Age	salary
101	a	xyz	23	20000
102	b	abc	43	56000
103	c	pqr	34	40000
104	d	asd	56	59000
105	e	abc	33	39000
106	f	def	27	29000
107	g	ghi	47	49000
108	h	JKL	55	70000
109	i	mno	28	30000
110	j	lop	38	36000

111	k	stu	49	50000
112	l	vwx	29	31000
113	m	yz	30	34698
114	n	qwe	40	49875
115	o	rtyu	50	58000
116	p	fngd	39	39785
117	q	sfh	25	26000
118	r	jkg	26	27154
119	s	mnb	36	36987
120	t	afgh	47	46987

Lab 3- Aggregate Transformation

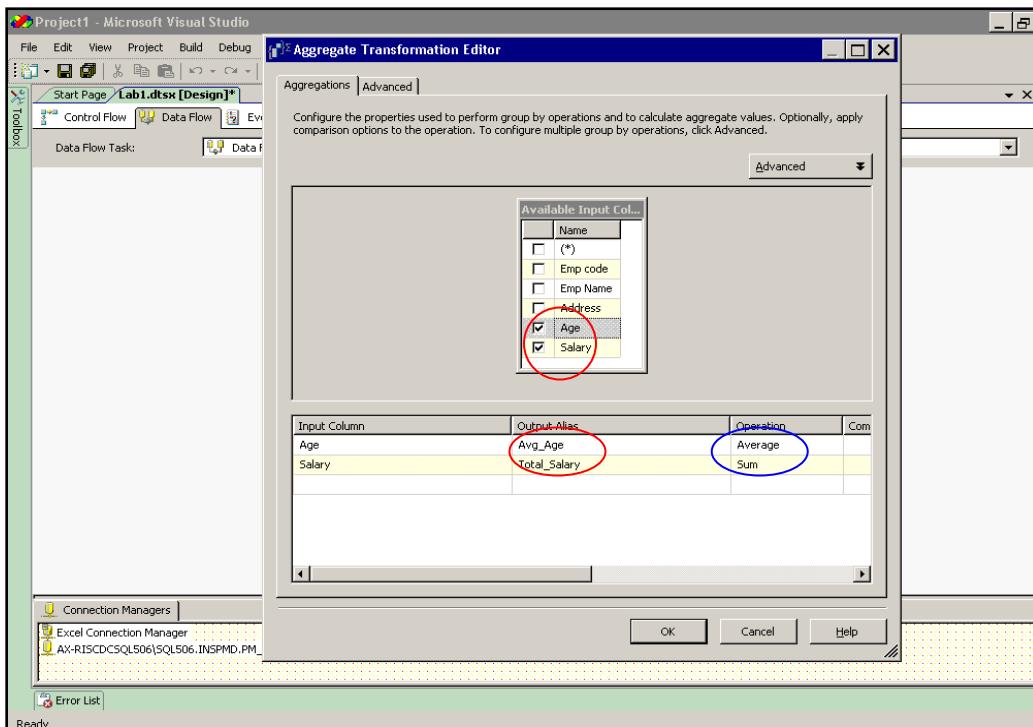
Objective	To find the average age in the organization and the total salary paid
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS Project and Packgae

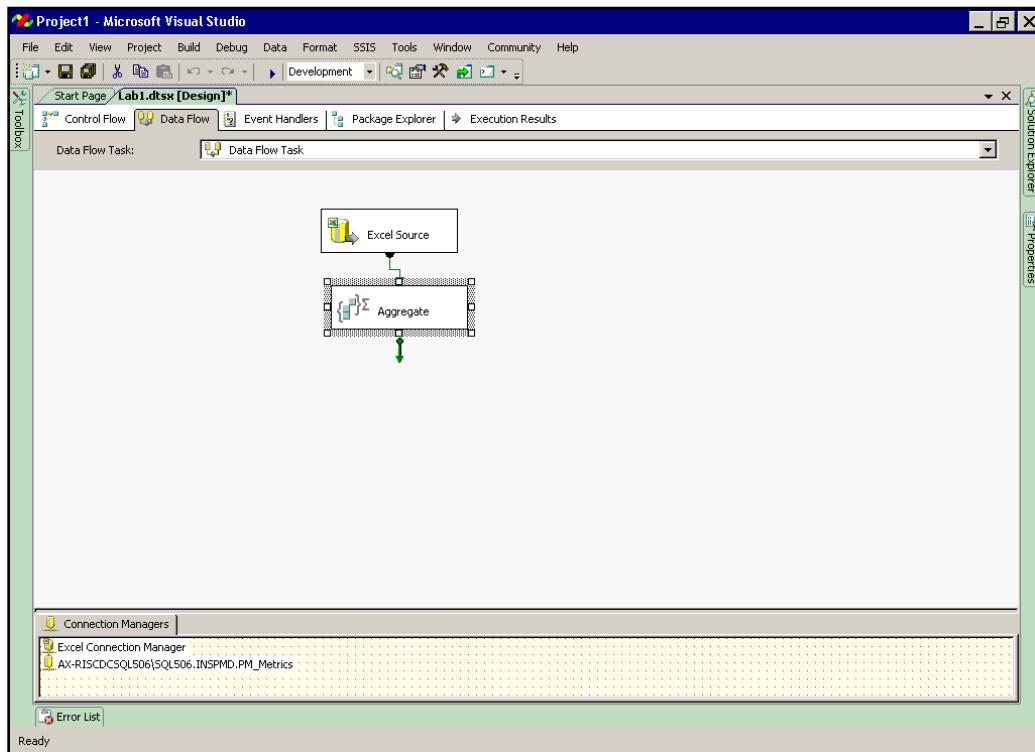
- 1) In the Data flow Task after selecting the source table, select aggregate transformation for the tool box.



2) Double click the transformation to open the editor.

- Select age and salary.
- Select the 'Operation' on AGE as average and on salary as SUM and then click on OK.

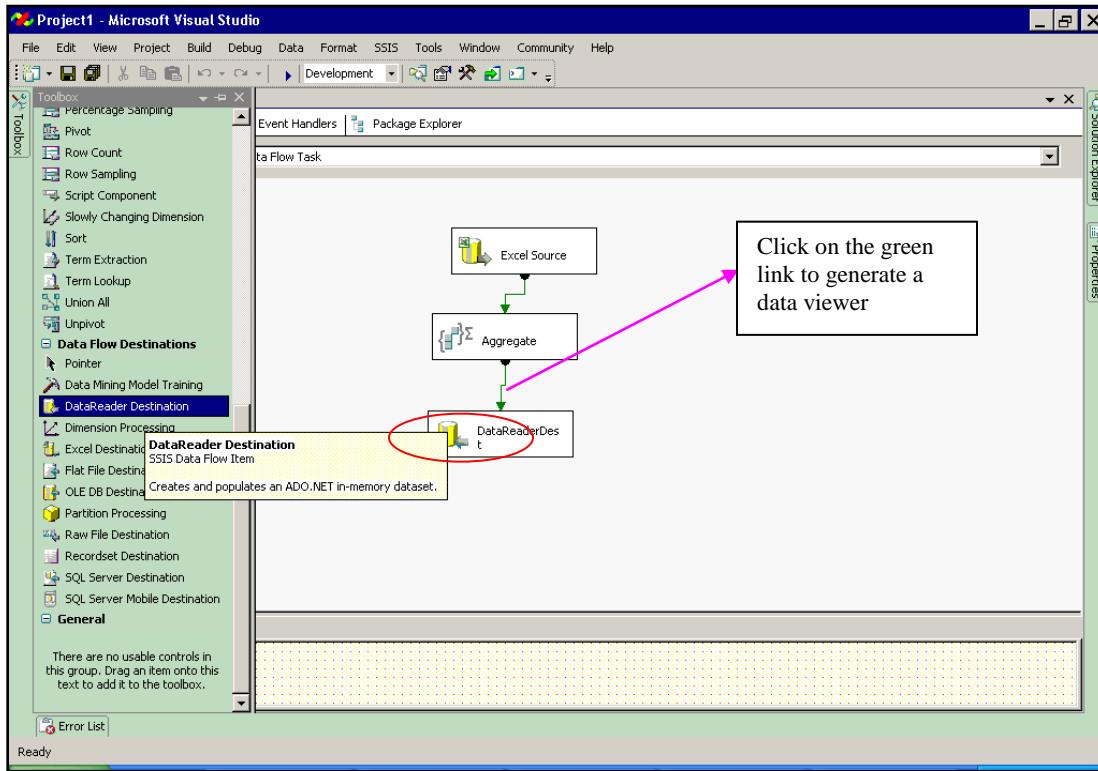




Notes:

You can use DATA READER Destination as dummy target table. This table is not save anywhere. It is only used to check the final data when we do not want to make any change or load the final data into the target table.

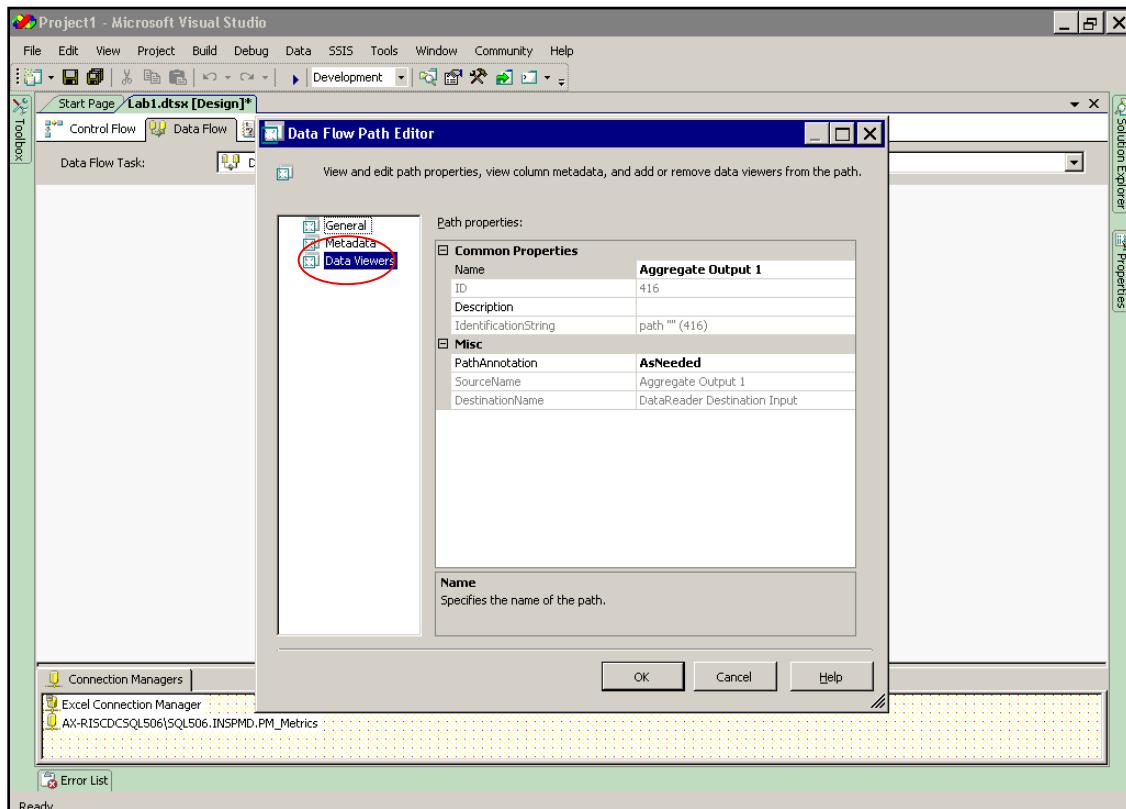
This destination can be used without any configuration changes, in effect creating a 'null' destination. The data goes nowhere but we can trouble shoot any issue in our derived column upstream before



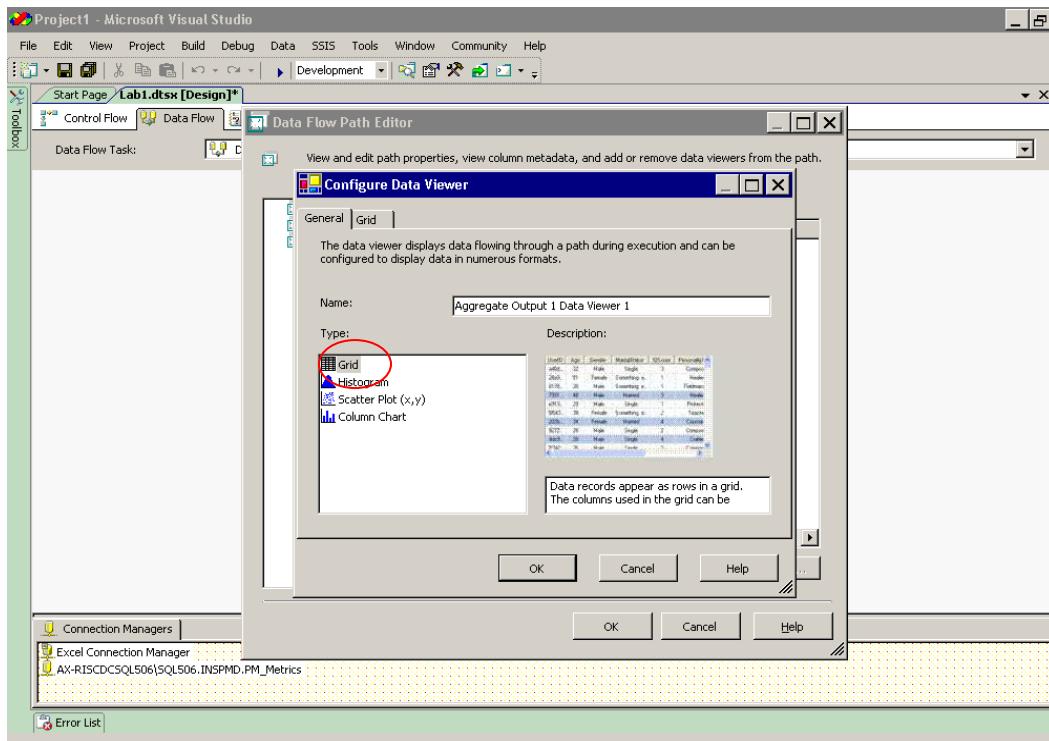
To view the data we have to use DATA VIEWER.

3) Select DATA READER Destination from the toolbox.

- Link the aggregate transformation to the DATA READER Destination.
- Double click the link to view the data flow editor and select Data Viewers.



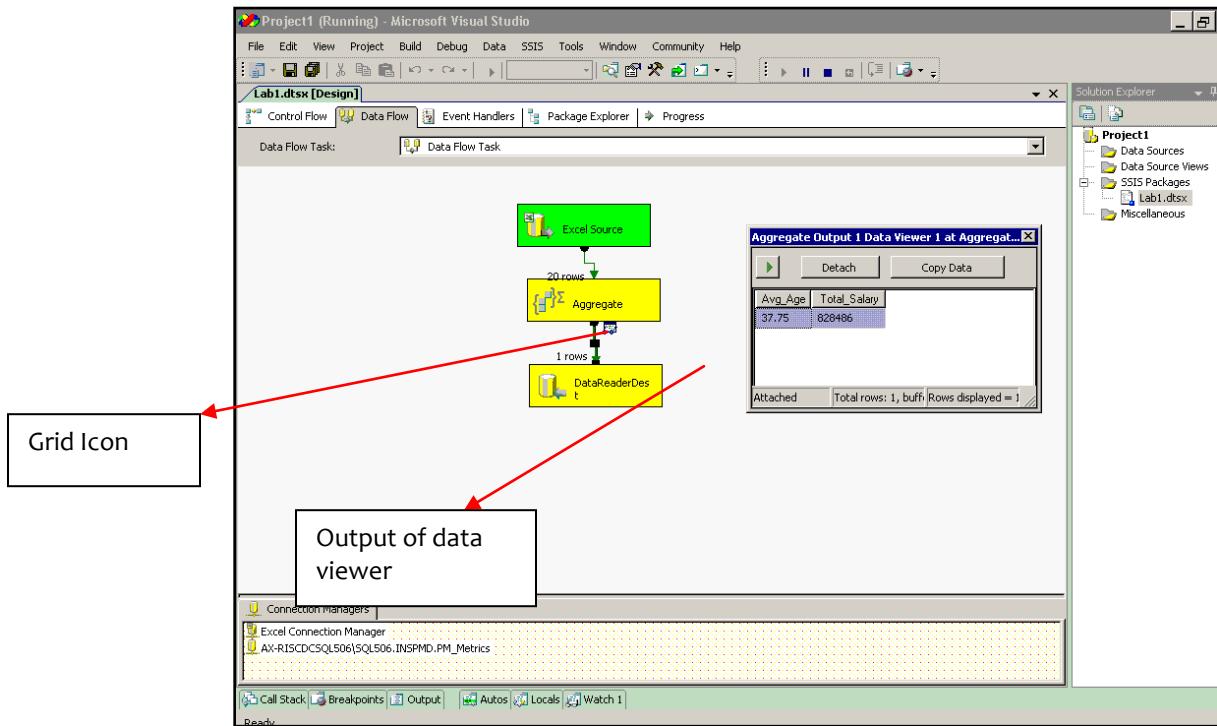
→ Add a new data viewer and select Grid and click OK twice.



4)A small icon of GRID will be visible near the joining arrow.

5)Save and execute the DATA FLOW TASK.

6)On successful execution a grid with output will be shown as:

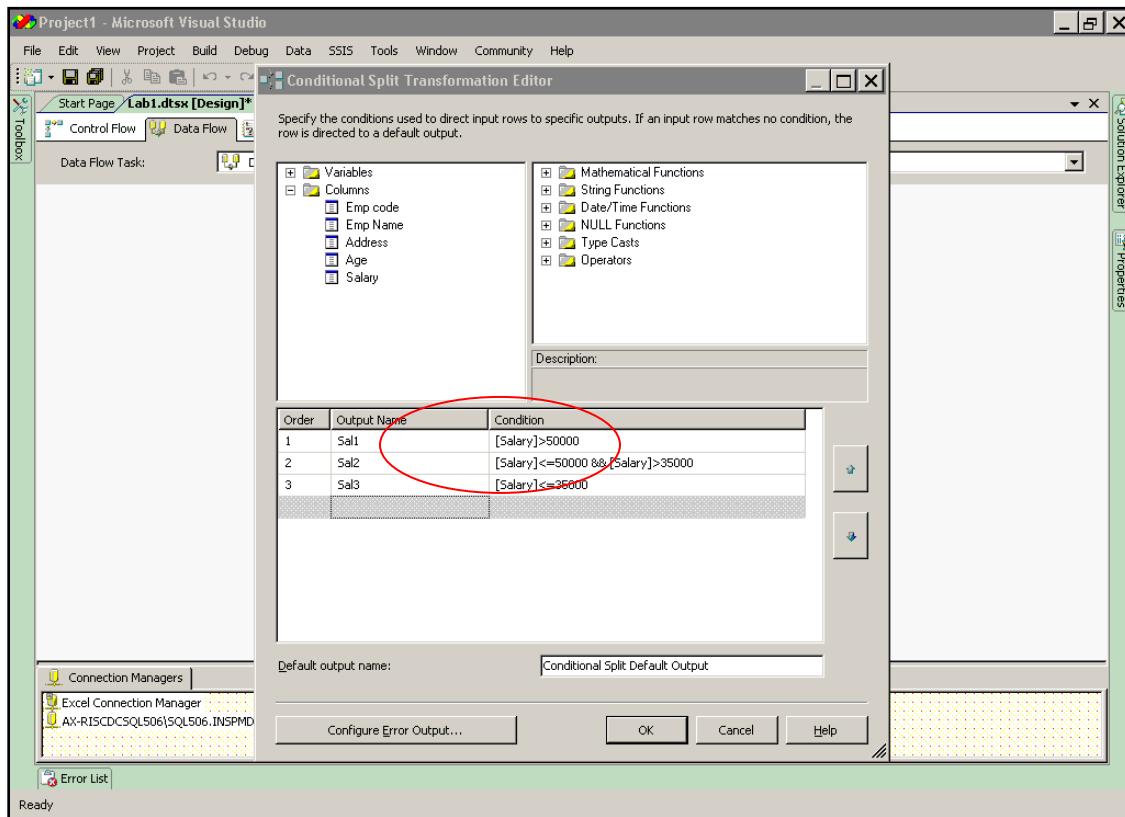


7)As soon as the data viewer is closed the execution gets completed.

Lab 4- Conditional split transformation

Objective	To load different tables with specific salary range (>50000, >35000 and <50000, <=35000) using conditional split and derived column transformation
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS project and Package

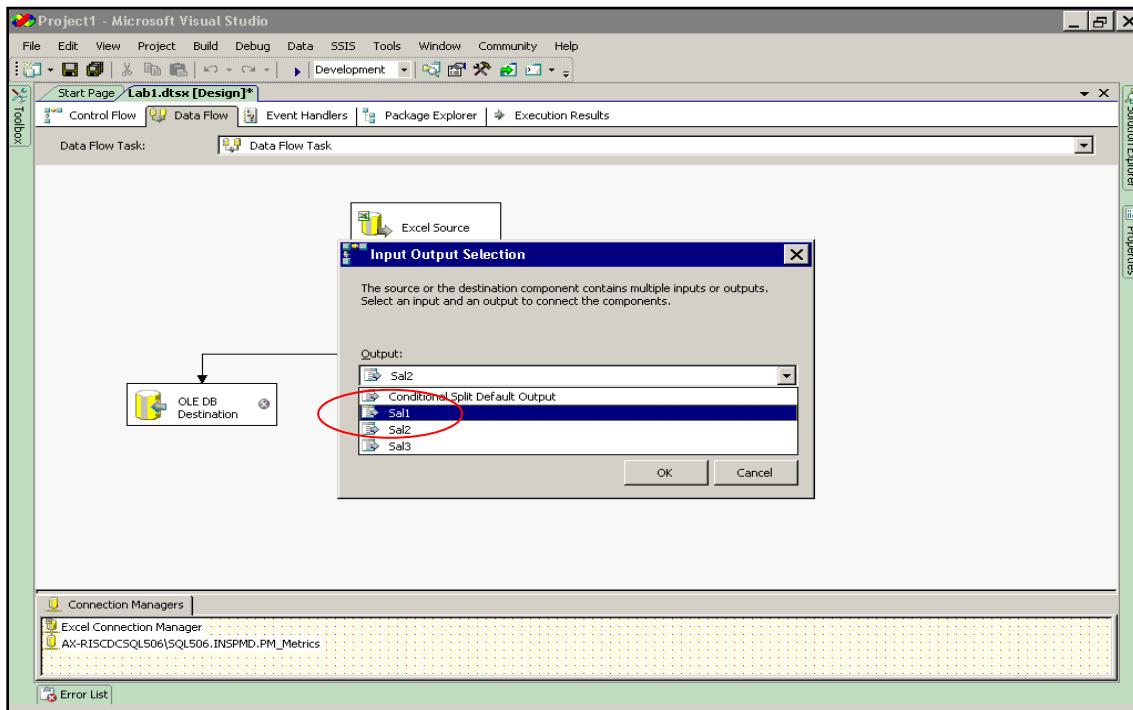
1)In the Data flow Task after selecting the source table, select conditional split transformation from the tool box and double click to get the editor.



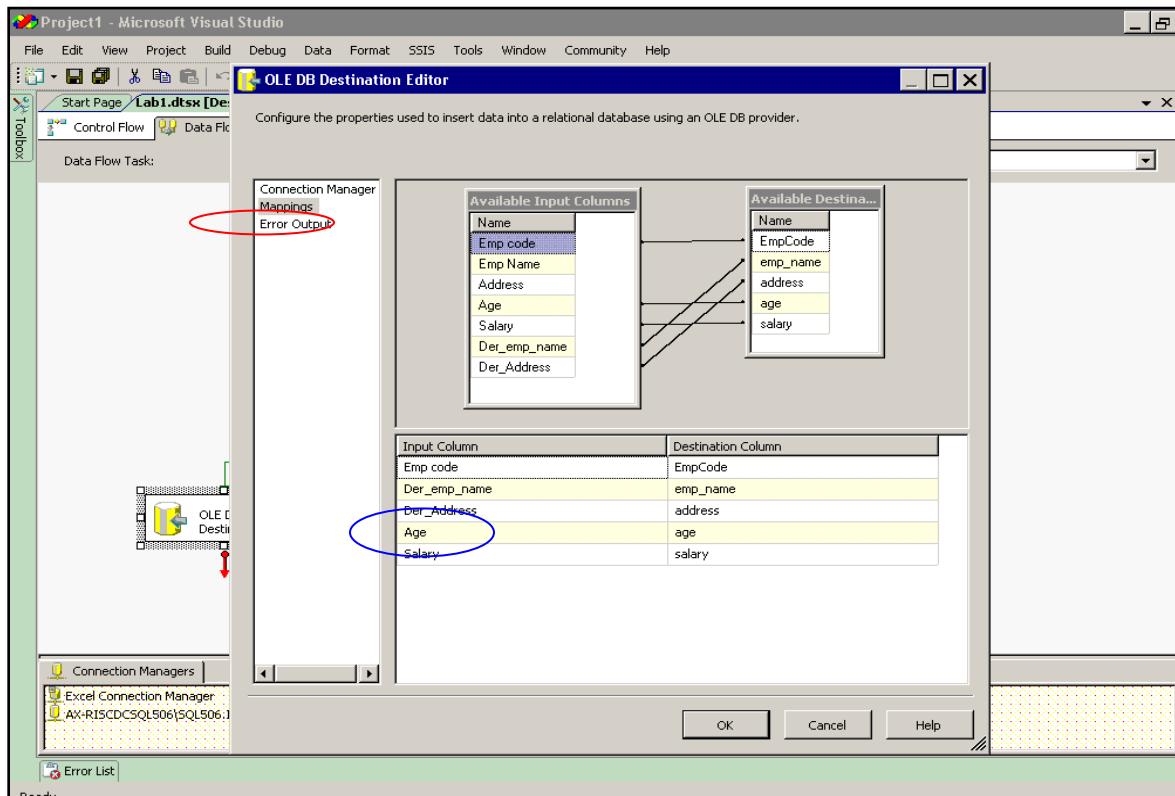
2)In the editor use different conditions according to which you want to divide the data.

3)Insert a Derived column transformation between source and conditional split and change the data type to name and address.

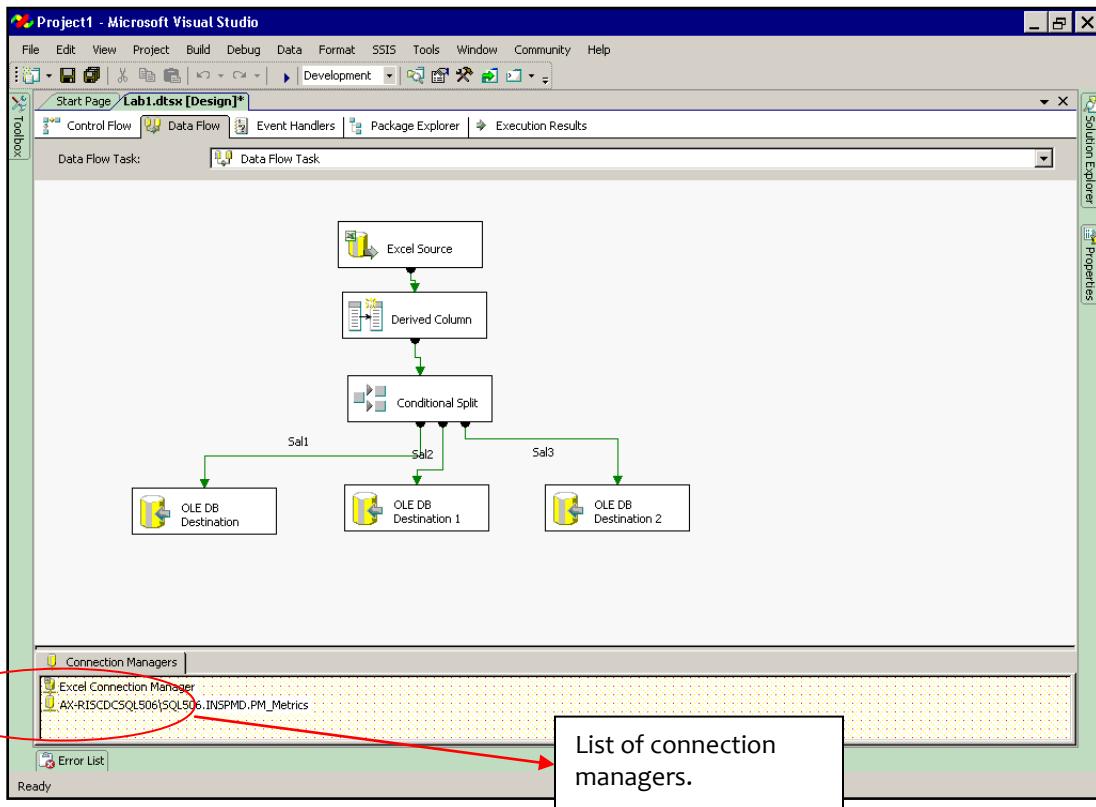
- ➔ Select target tables (OLE DB) and join with the transformation (select valid output).



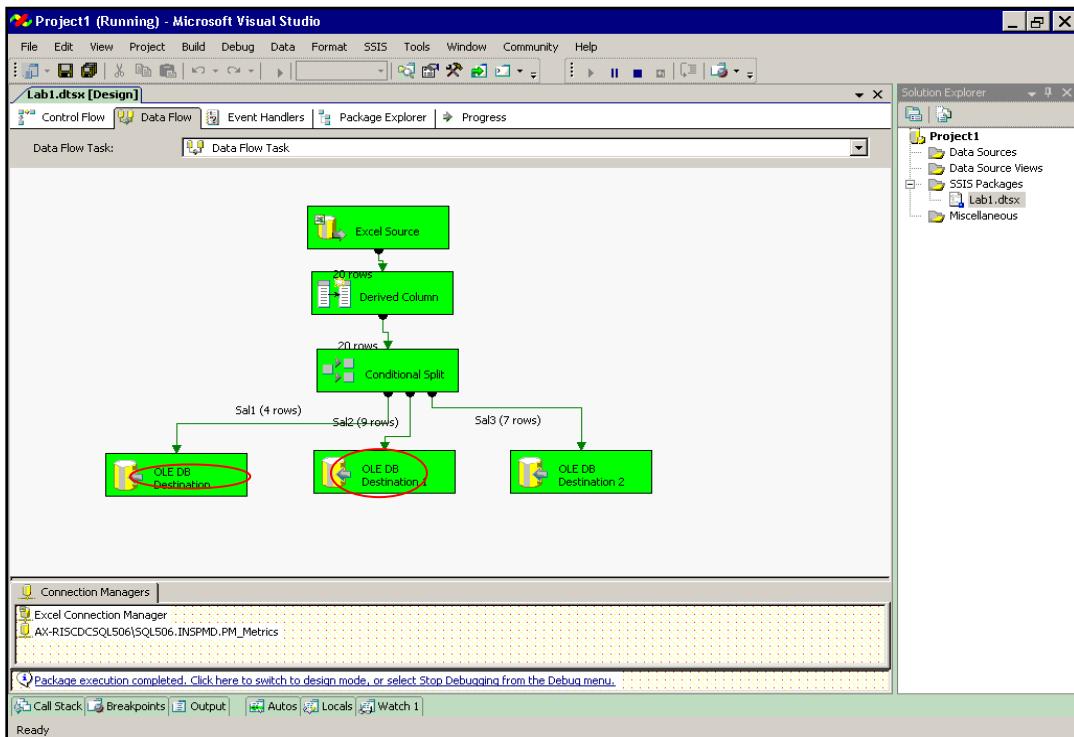
- You can use the existing connection manager or can create a new connection manager for the target table.
- Use DATA ACCES MODE as –Table or view.
- Select appropriate mapping.



4) Repeat for all the target tables.



5) Save and execute.



6)Using “SELECT” statement in SQL server.

Destination 1 :

Emp_code	emp_name	Address	Age	salary
102	b	abc	43	56000
104	d	asd	56	59000
108	h	jk	55	70000
115	o	rtyu	50	58000

Destination 2 :

Emp_code	emp_name	Address	Age	salary
103	c	pqr	34	40000
105	e	abc	33	39000
107	g	ghi	47	49000
110	j	lop	38	36000
111	k	stu	49	50000
114	n	qwe	40	49875
116	p	fnfd	39	39785
119	s	mnb	36	36987
120	t	afgh	47	46987

Destination 3

Emp_code	emp_name	Address	Age	salary
101	a	xyz	23	20000
106	f	def	27	29000
109	i	mno	28	30000
112	l	vwx	29	31000
113	m	yz	30	34698
117	q	sfh	25	26000
118	r	jkg	26	27154

Lab 5- Look Up Transformation

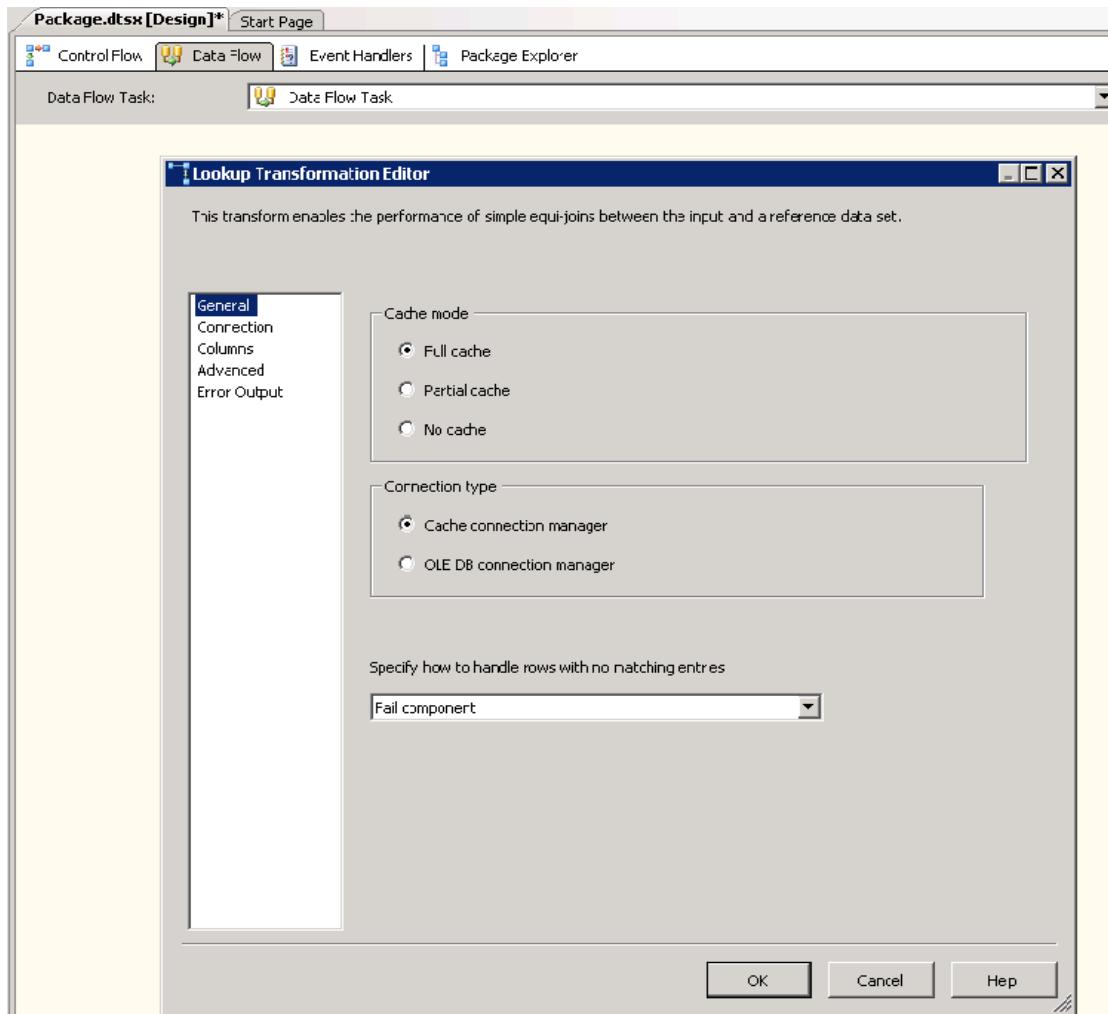
Objective	To check in the database whether there is any change in the address of the employee or if any new employee details are added then load the data in the database.
Lab Setup	<ul style="list-style-type: none">• SSDT tool• Existing SSIS project and Package

1)In the Data flow Task after selecting the source table, drag and drop Data conversion transformation and double click to view the editor.

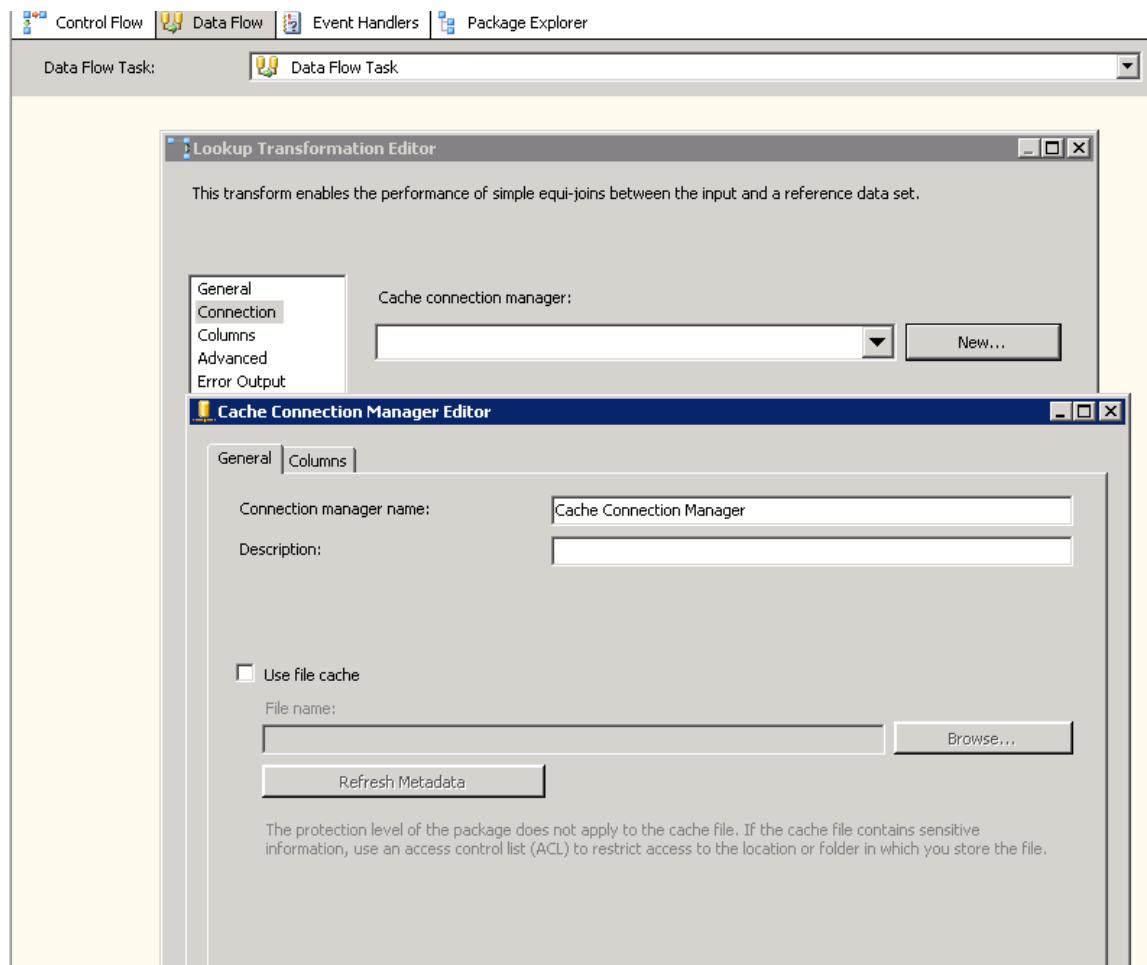
→ Convert the data type of the fields like emp_name, emp_code, address and salary compatible with the target table.

2)Drag and drop the Look up transformation and then double click to view the editor.

3) In the General tab, you can select the Cache Mode and the Connection Type:



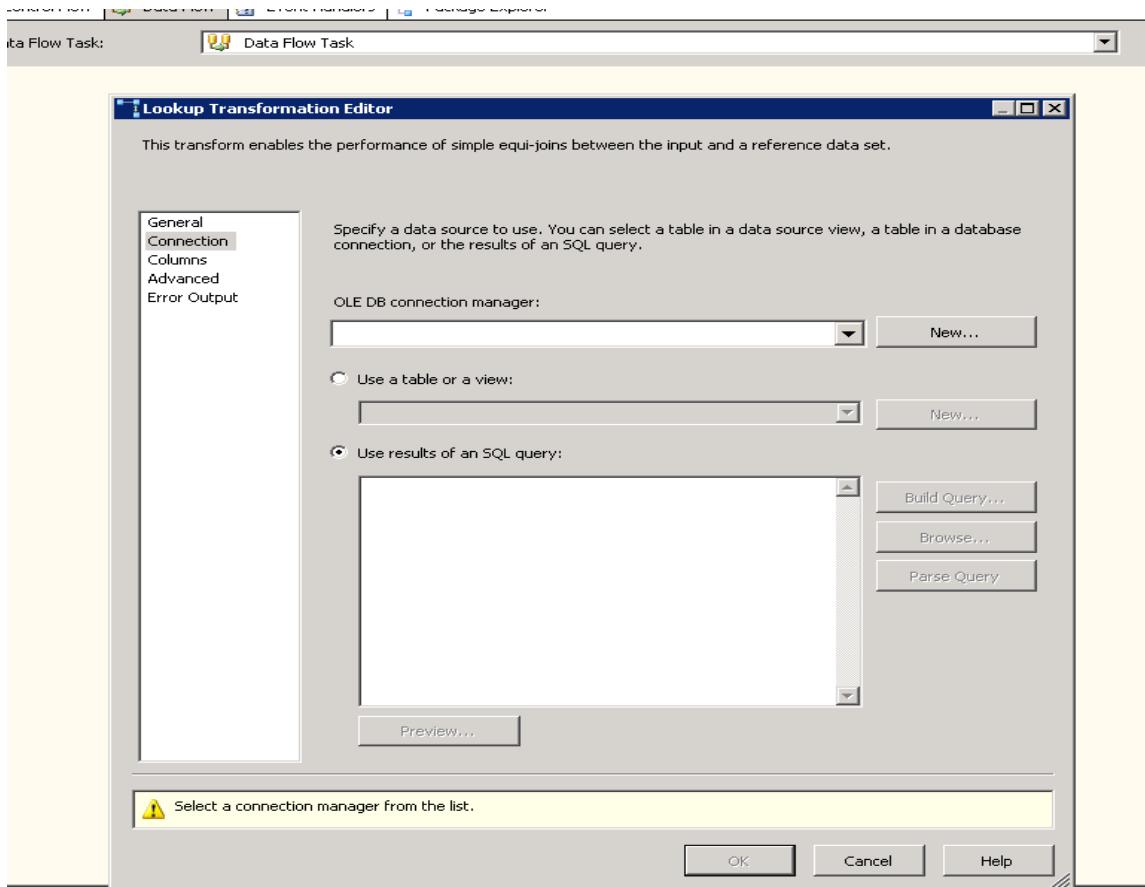
4) After selecting Cache Connection Manager in the Connection Type, You can select the Cache Connection Manager which you have created in the Cache Transformation created before the Lookup Transformation.



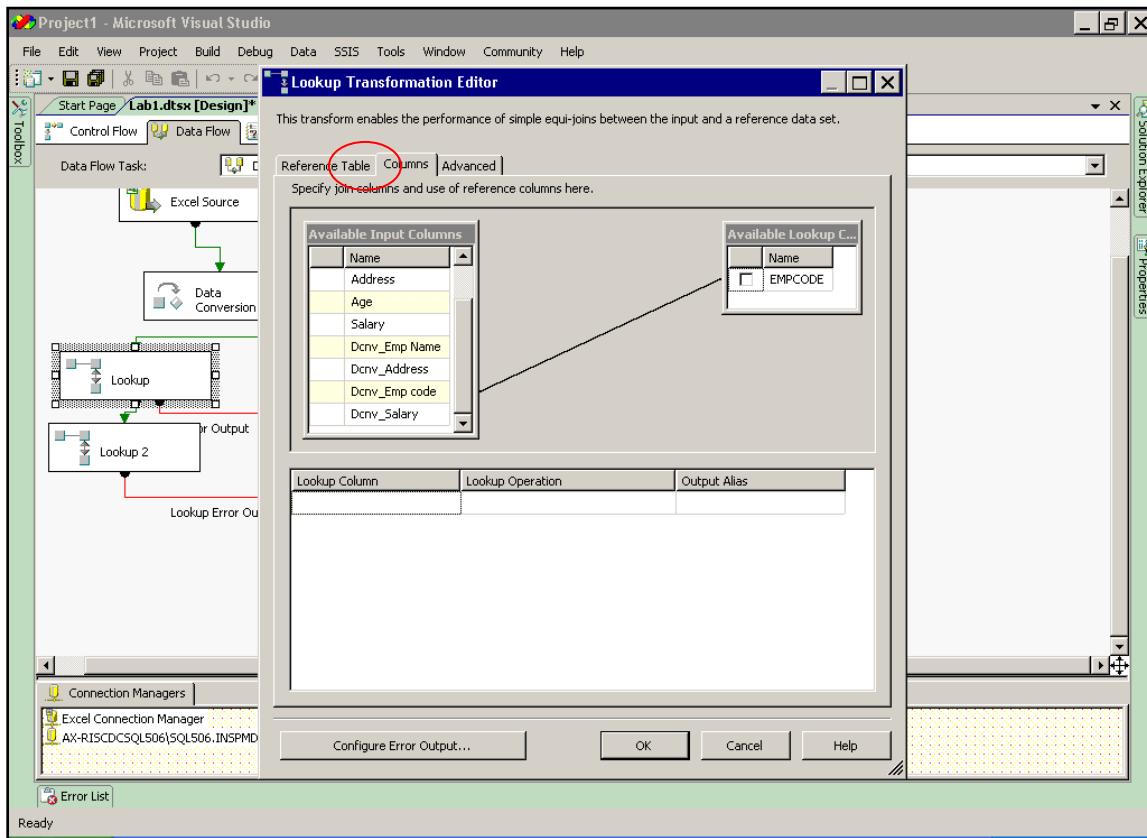
5) After selecting OLE DB Connection Manager in the Connection Type, In the editor you can either use a table or use SQL statement to look up the data.

6) Select the connection manager and use a SQL statement to extract empcode.

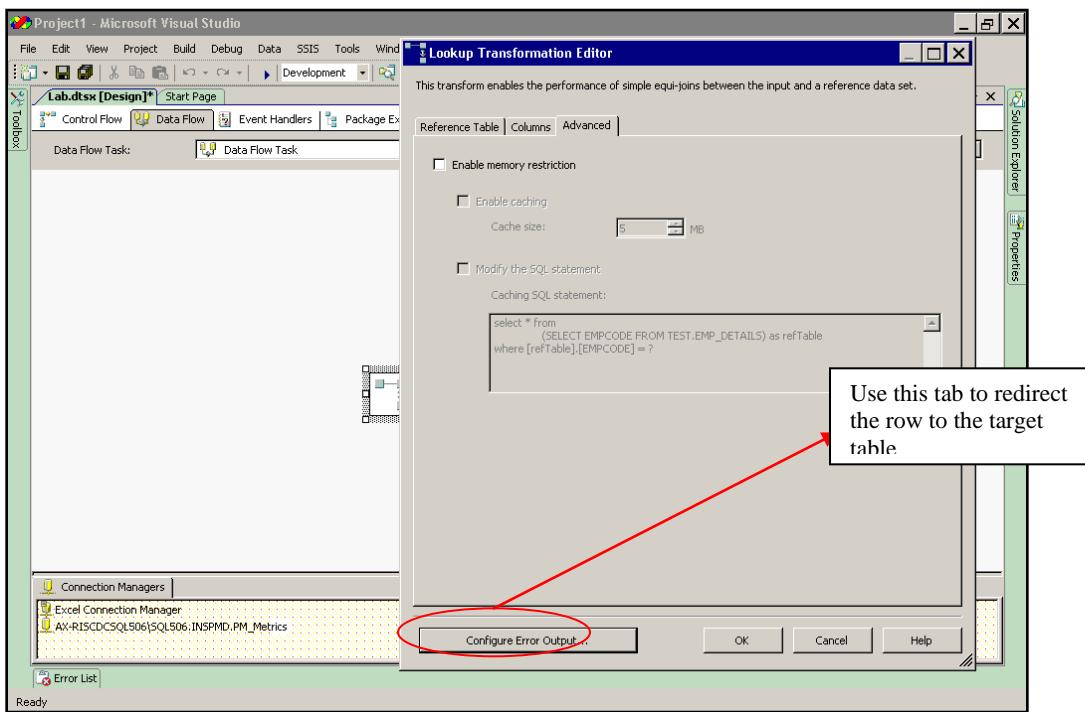
(Select empcode from table name)

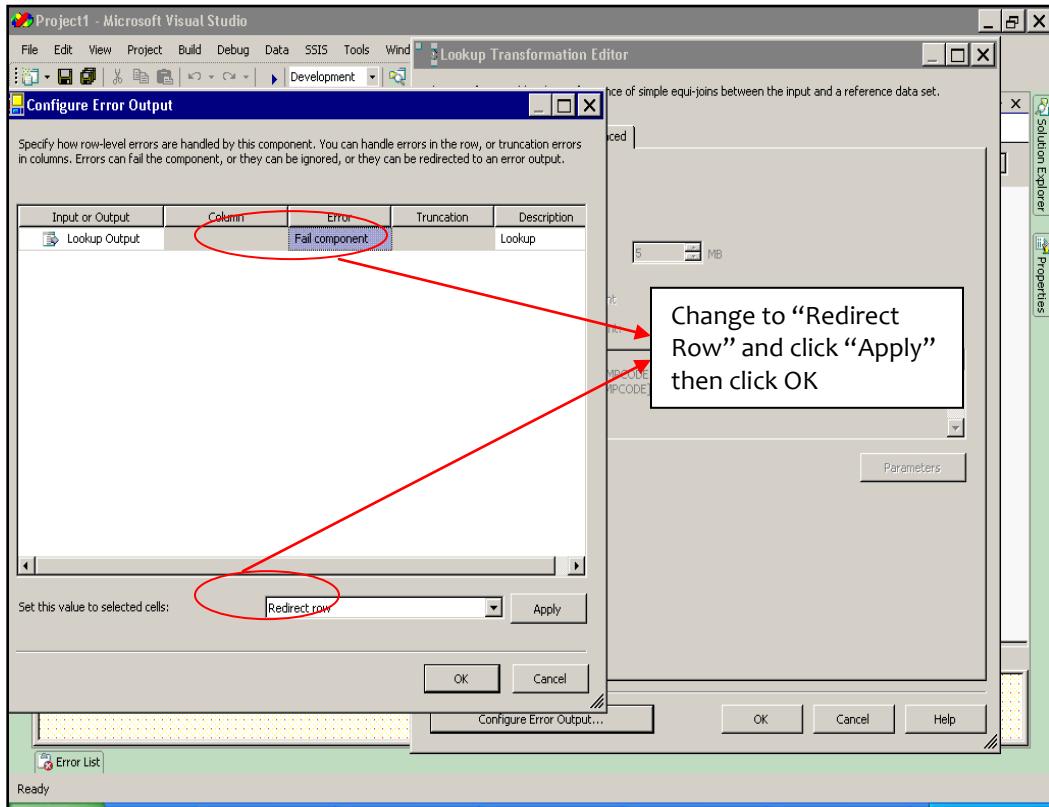


- 7) On the column tab join the empcode to find a match for the same.

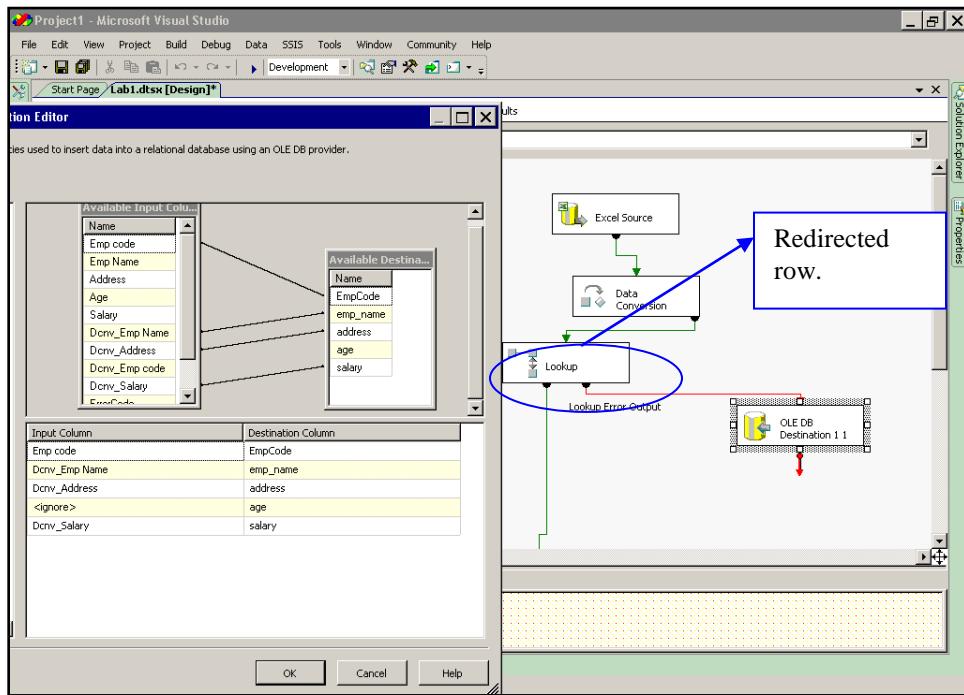


- 8) If the lookup finds any new data then it redirects to the target table.





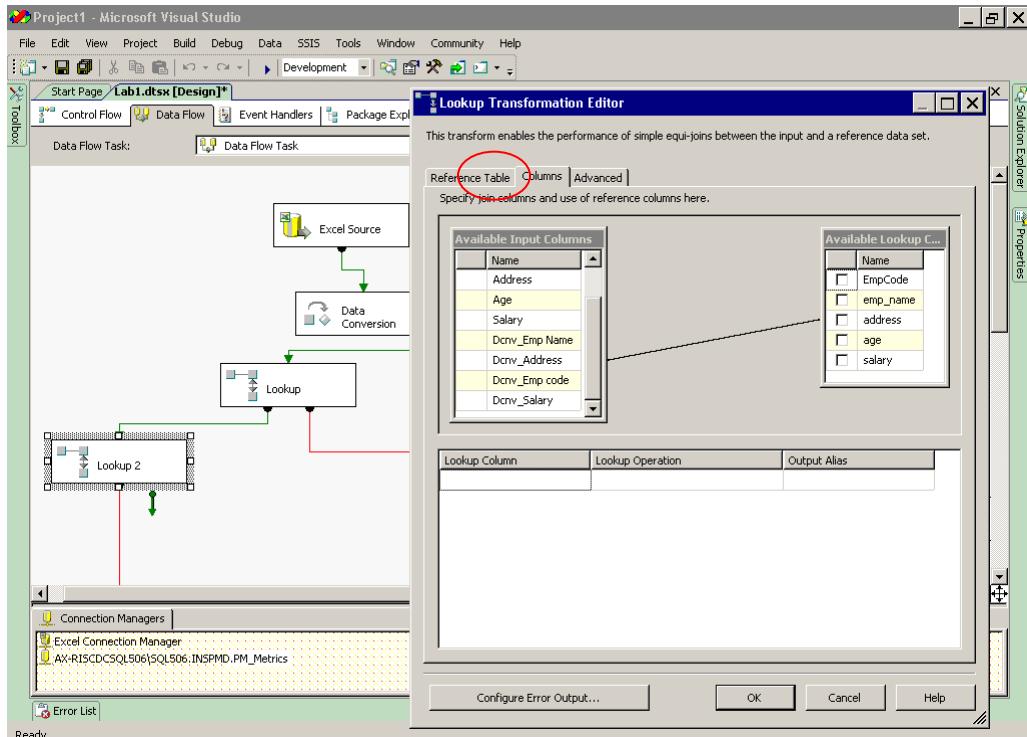
9) Use one OLE DB destination to redirect the new row to target table. In the OLE DB destination set the connection properties and map the columns.



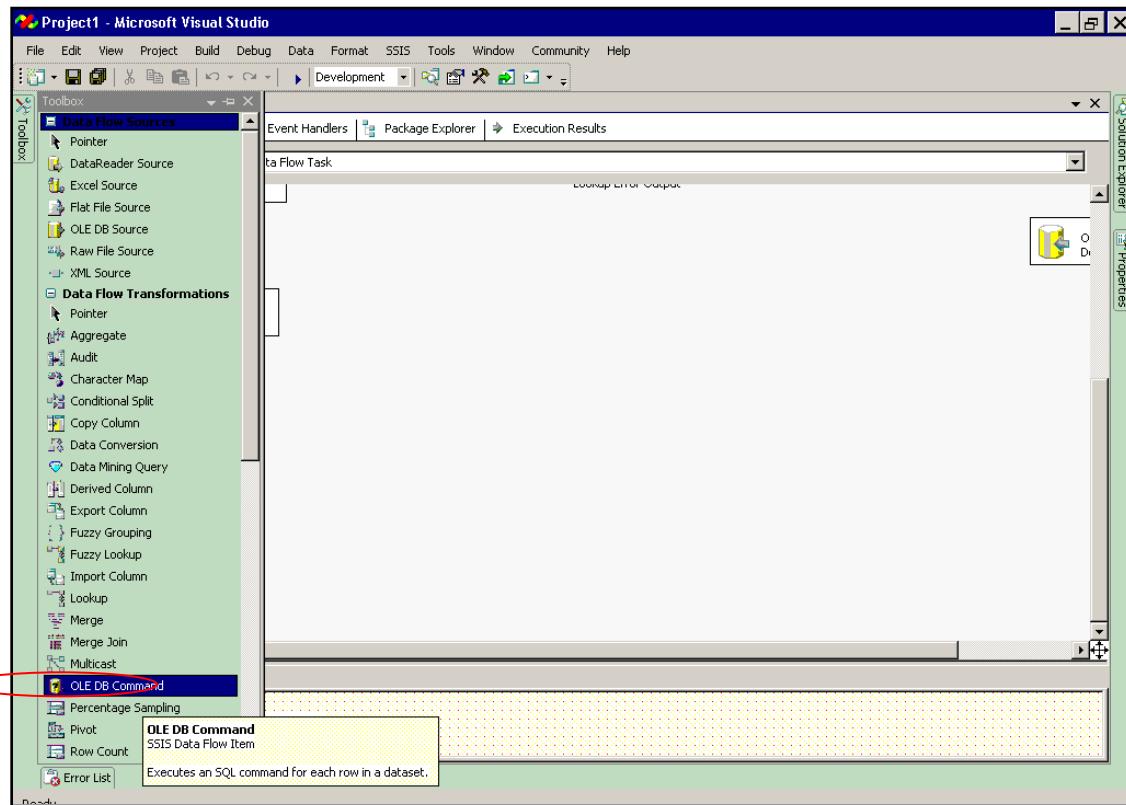
10) Use another look up to check if there is any change in address of the data that is already present in the target table.

→ To use the result of an SQL query use the following query:
SELECT * FROM TEST.EMP_DETAILS

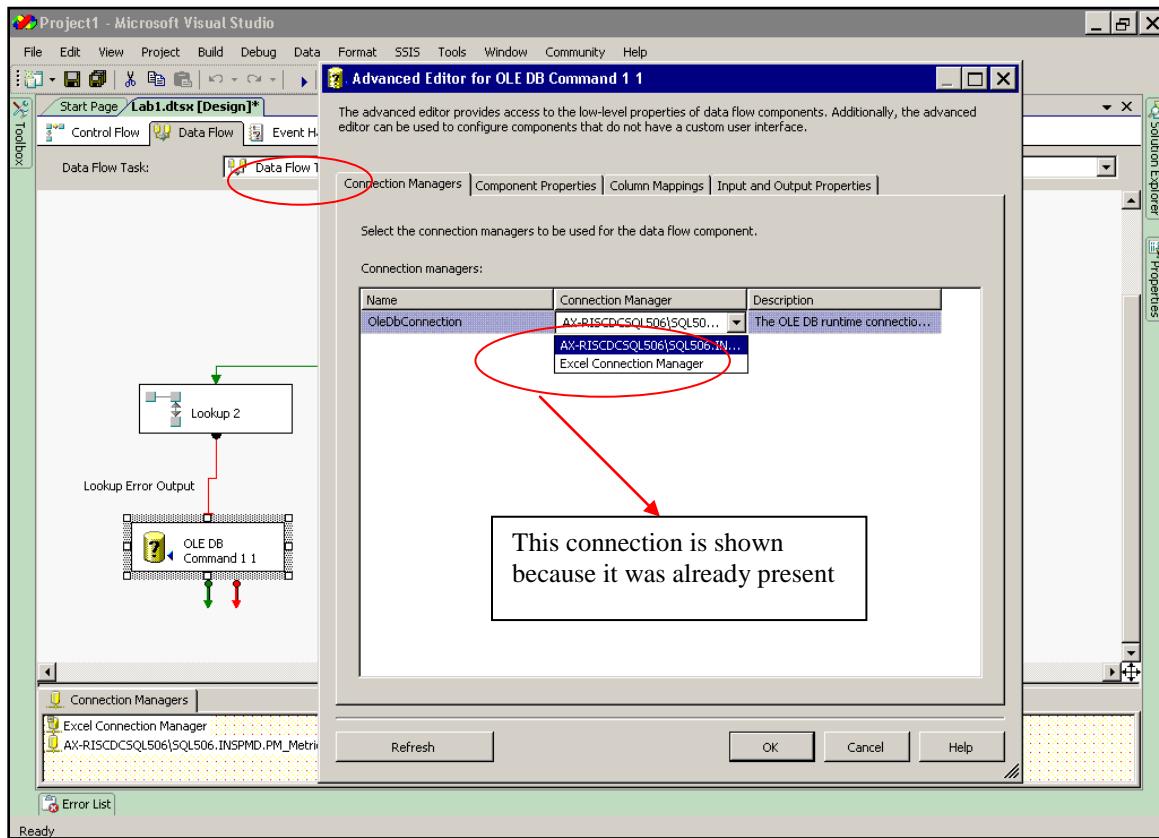
11) In the Columns tab match the address columns to check.



- 12) Now select OLE DB command from the toolbox.



13) Set the connection as available:

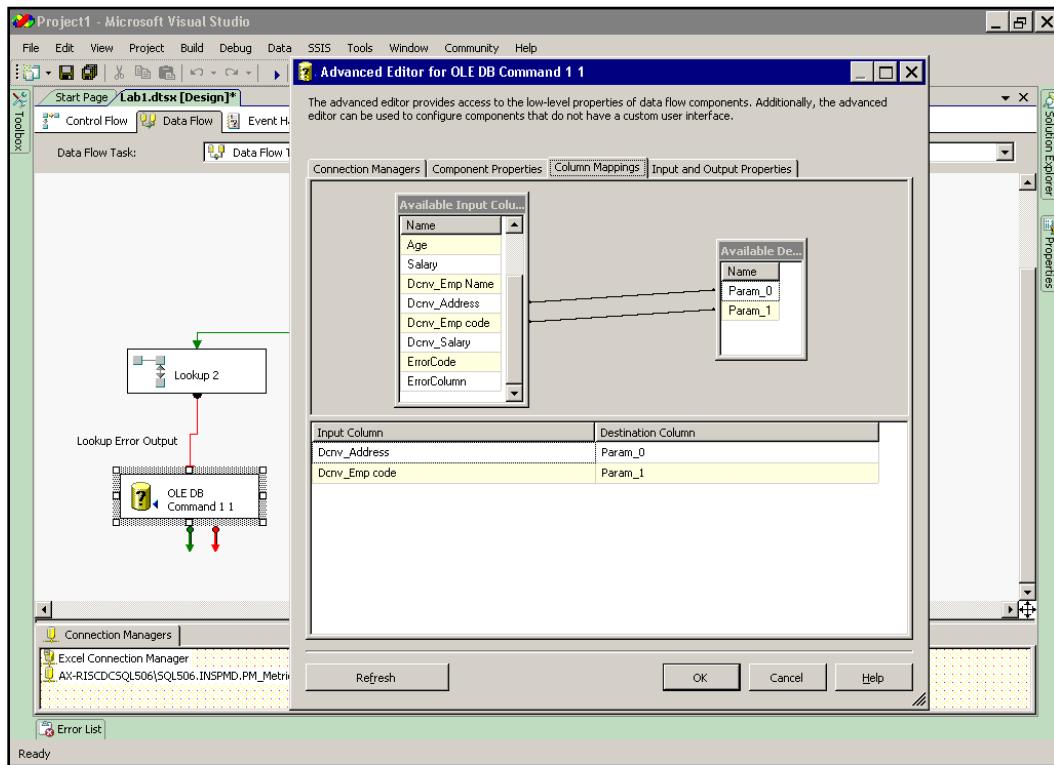


→ In the component properties tab select the SqlCommand and write the query as:

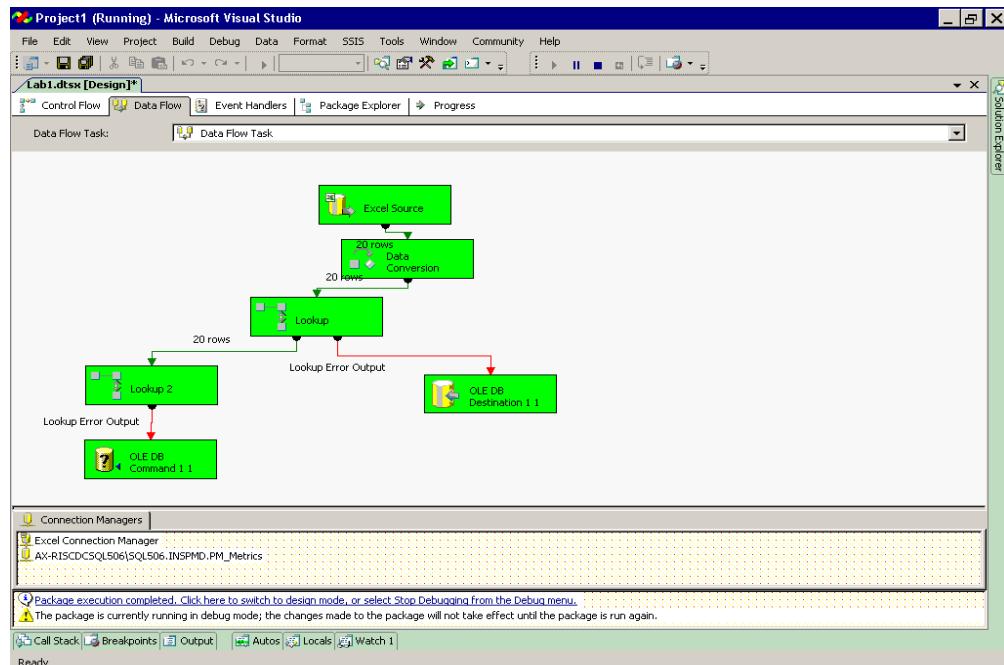
```
UPDATE TEST.EMP_DETAILS
SET ADDRESS=?
WHERE EMPCODE=?
```

[The two question marks create parameters which are used to check the data (Old and updated).
The first question mark refer to first and second refer to the second param]

14) In the column Mappings tab map the columns with the parameters:



15) Now save and execute the Data flow.



Original data:

Emp_code	emp_name	Address	Age	salary
101	a	xyz	23	20000
102	b	abc	43	56000
103	c	pqr	34	40000
104	d	asd	56	59000
105	e	abc	33	39000
106	f	def	27	36254
107	g	ghi	47	49000
108	h	jkl	55	70000
109	i	mno	28	30000
110	j	lop	38	36000
111	k	stu	49	50000
112	l	vwx	29	31000
113	m	yz	30	32000
114	n	qwe	40	49875
115	o	rtyu	50	58000
116	p	fnfd	39	39785
117	q	sfh	25	26000
118	r	jkg	26	27154
119	s	mnb	36	35647
120	t	afgh	47	46987

Final data:

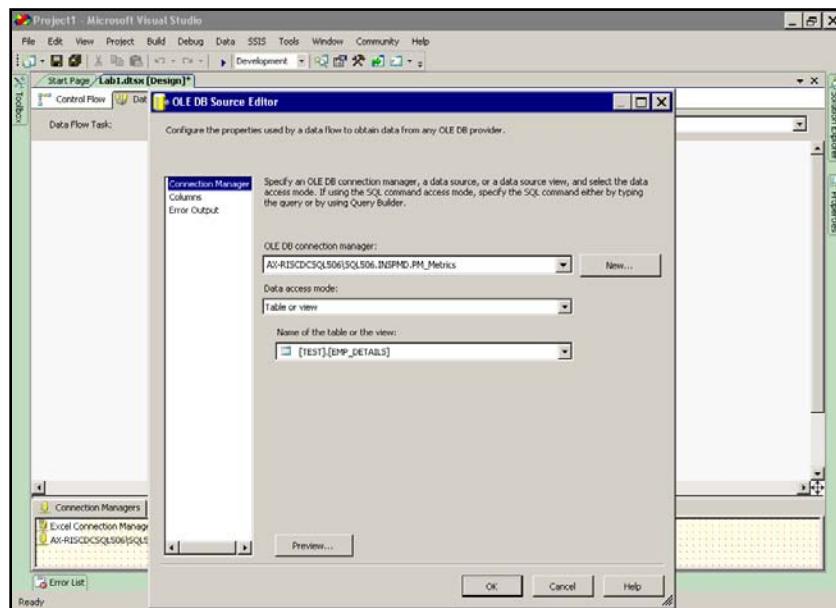
Emp_code	emp_name	Address	Age	salary
101	a	Klasfnx	23	20000
102	b	abc	43	56000
103	c	pqr	34	40000
104	d	asd	56	59000
105	e	abc	33	39000
106	f	perghjsd	27	36254
107	g	ghi	47	49000
108	h	jkl	55	70000
109	i	mno	28	30000
110	j	lop	38	36000
111	k	jshfkag	49	50000
112	l	vwx	29	31000
113	m	yz	30	32000

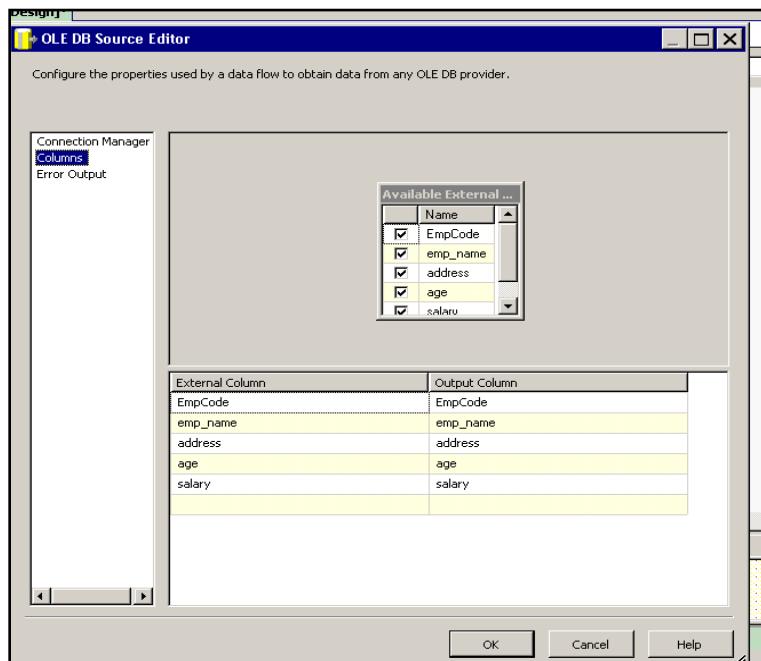
114	n	qwe	40	49875
115	o	rtyu	50	58000
116	p	fnfd	39	39785
117	q	sdfjfgosf	25	26000
118	r	jkg	26	27154
119	s	mnb	36	35647
120	t	afgh	47	46987

Lab 6- Copy Column Transformation

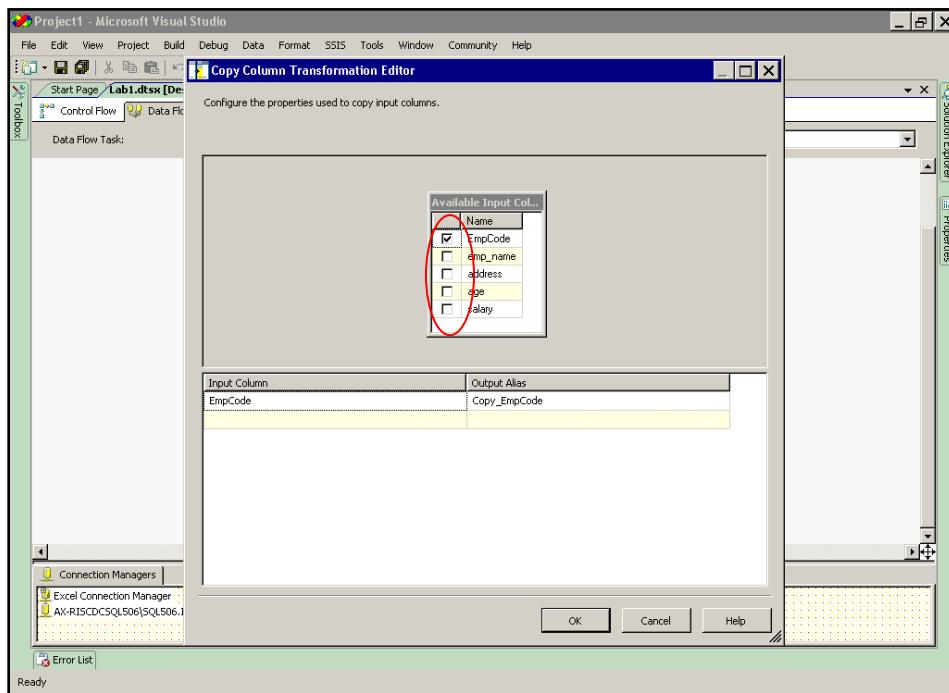
Objective	To create a copy of a column.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS project and Package

- 1)Select OLE DB source and double click to view the editor, set the connection manager and select the required columns.



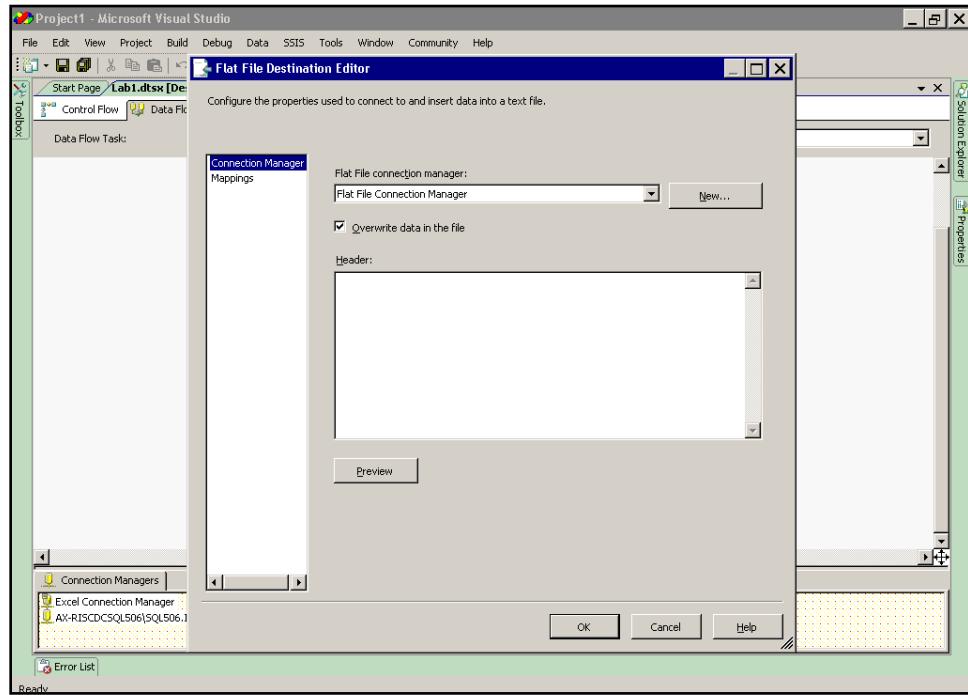


2) Select copy column transformation and double click to view the editor.

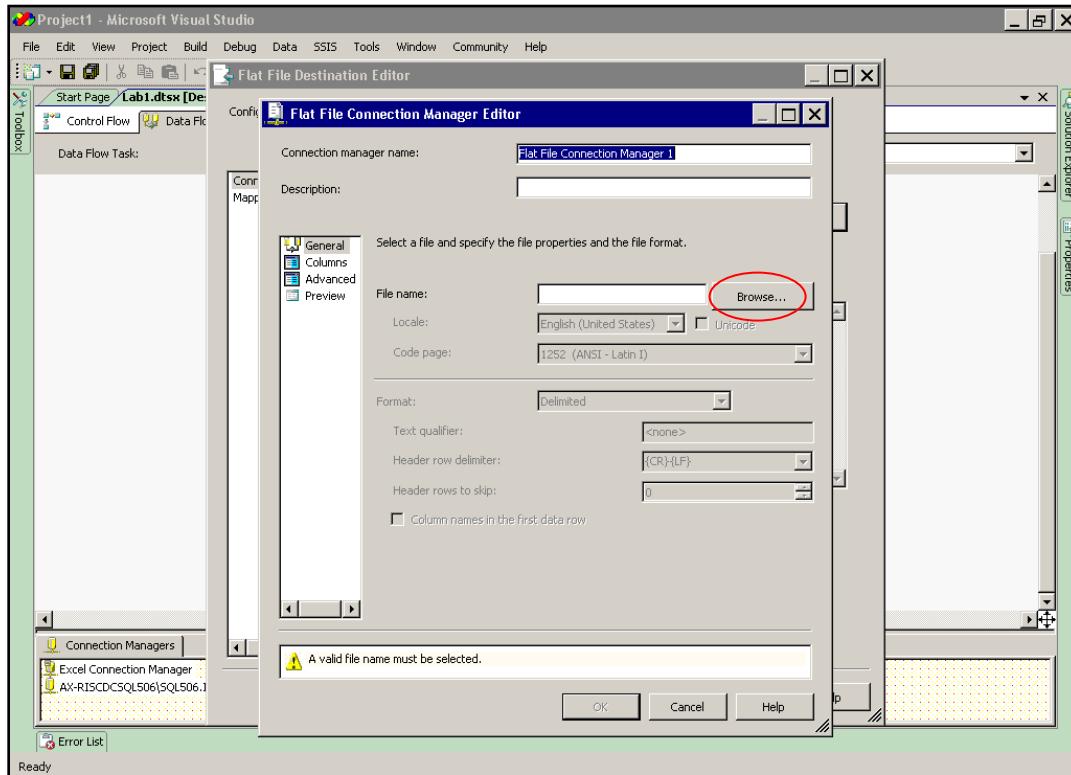


3)Select the column for which you need a copy and click OK.

4)Select a Flat file destination and double click to view editor.

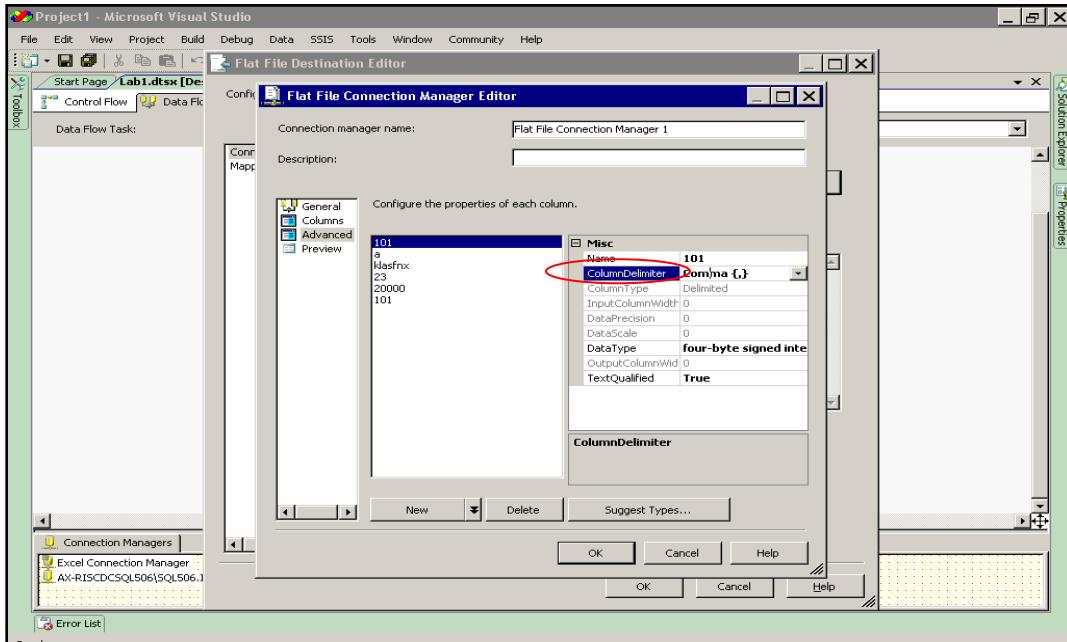


- 5) Click on NEW to get the format.
6) Select Delimited.

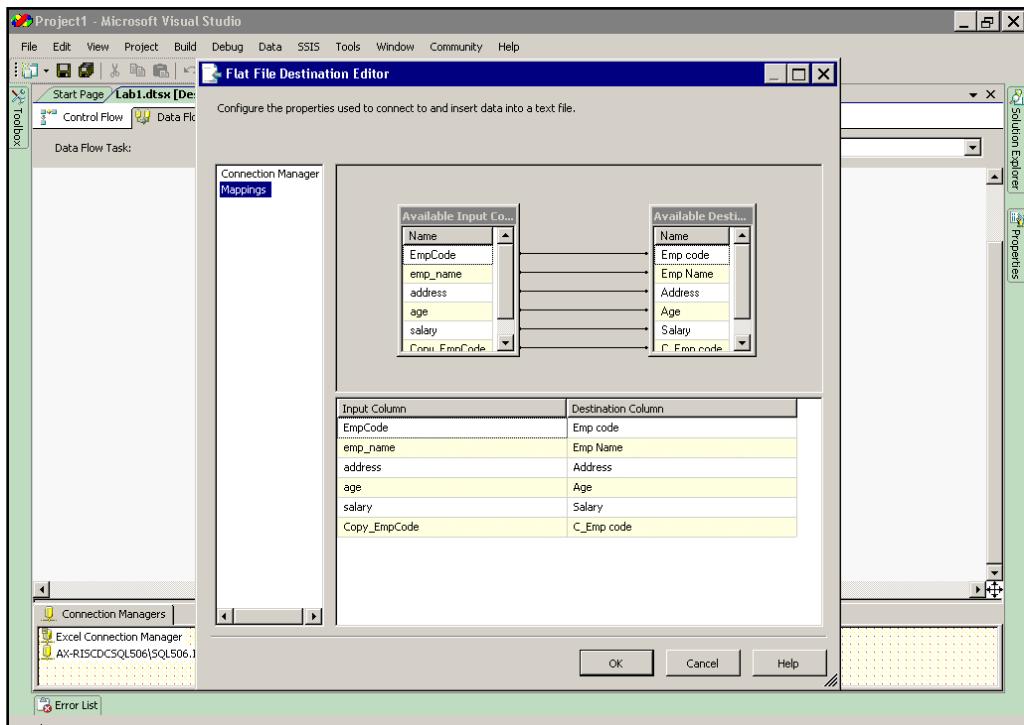


7)Browse for a file where you want to load the data.

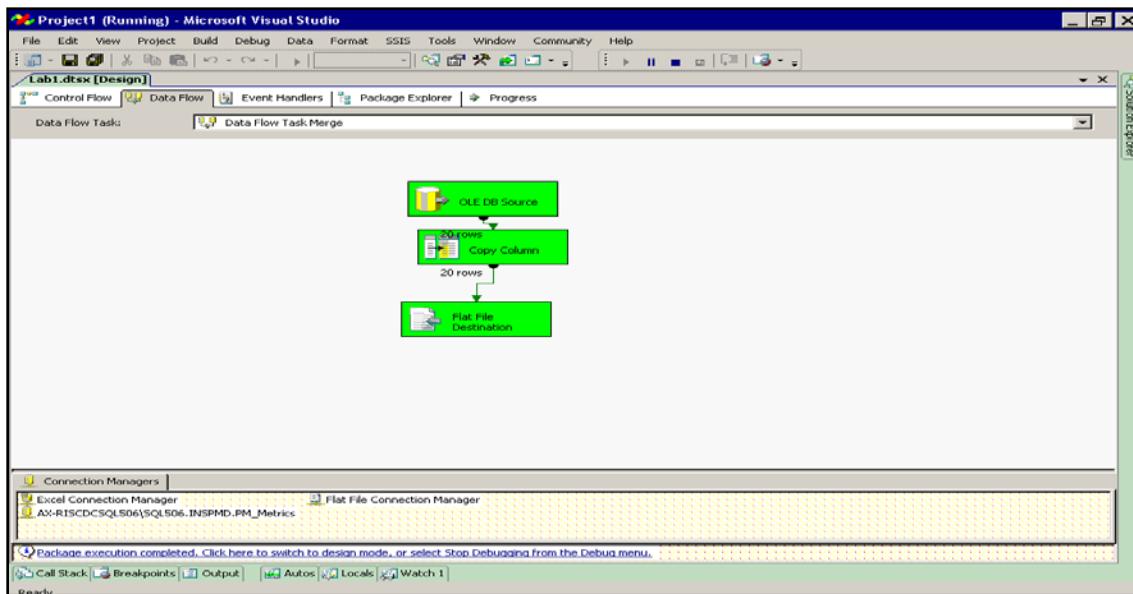
- Usually the column delimiter is Comma (,) which can be changed in Columns or advanced option.



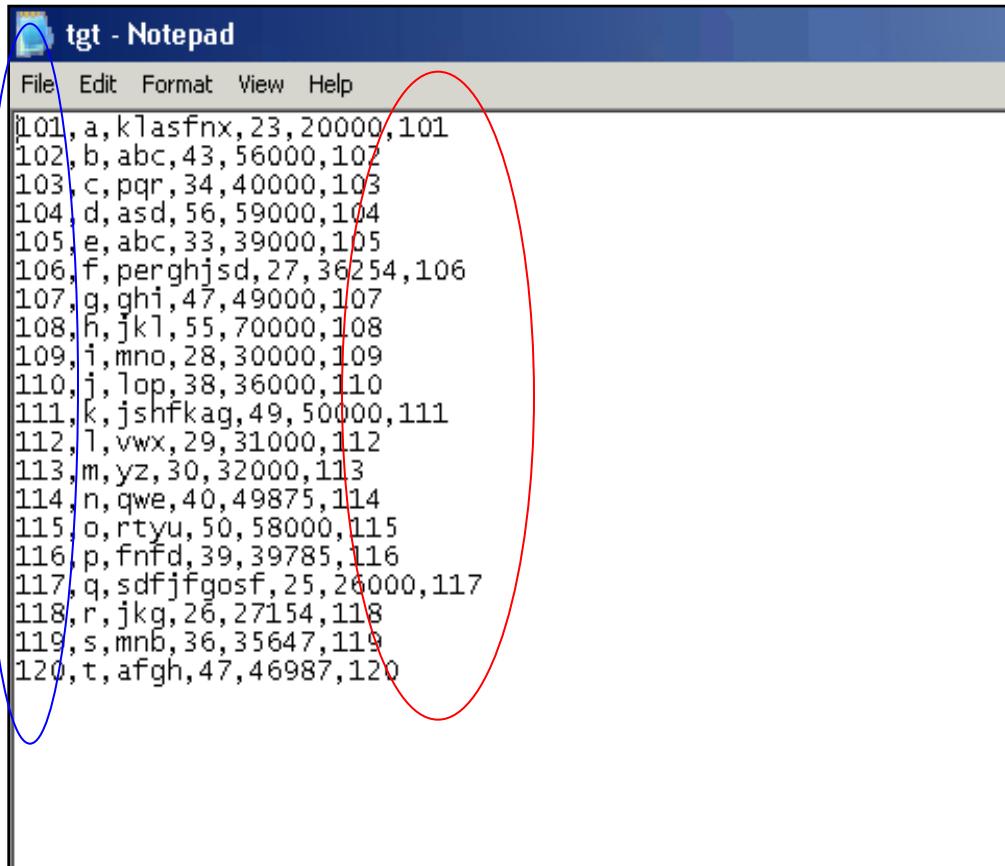
8)Select mapping and map the corresponding columns.



9) Save and execute the data flow.



Output:

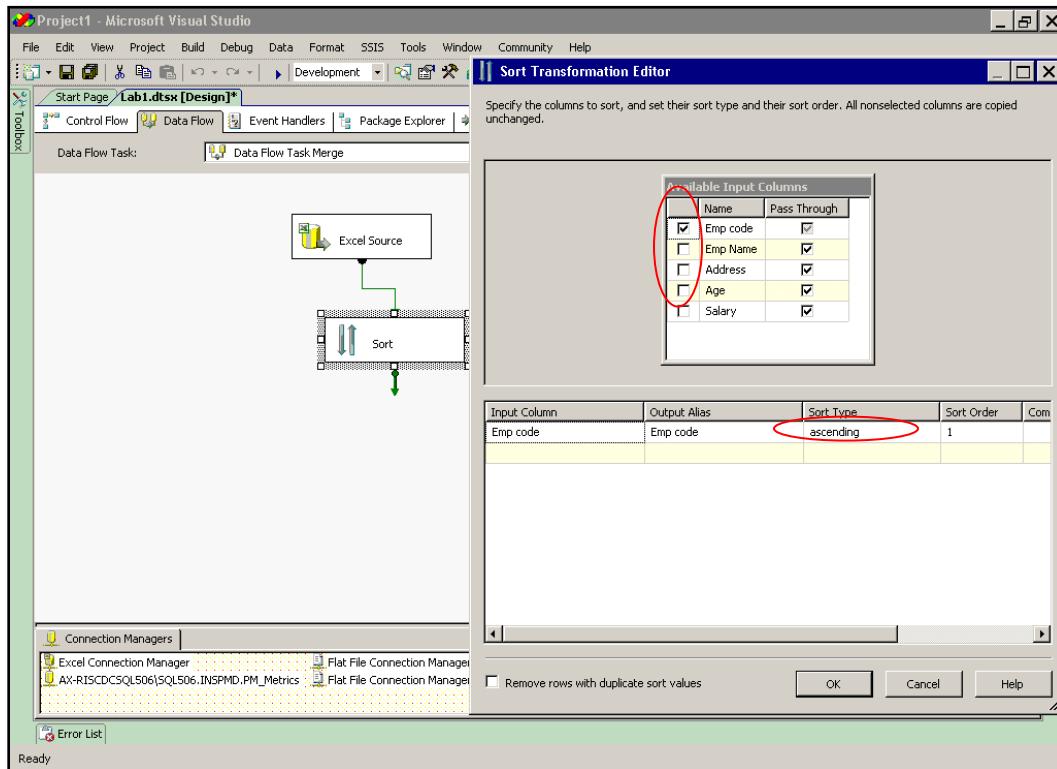


```
File Edit Format View Help
101,a,klasfnx,23,20000,101
102,b,abc,43,56000,102
103,c,pqr,34,40000,103
104,d,asd,56,59000,104
105,e,abc,33,39000,105
106,f,perghjsd,27,36254,106
107,g,ghi,47,49000,107
108,h,jkl,55,70000,108
109,i,mno,28,30000,109
110,j,lop,38,36000,110
111,k,jshfkag,49,50000,111
112,l,vwx,29,31000,112
113,m,yz,30,32000,113
114,n,qwe,40,49875,114
115,o,rtyu,50,58000,115
116,p,fmfd,39,39785,116
117,q,sdfjfgosf,25,26000,117
118,r,jkg,26,27154,118
119,s,mnb,36,35647,119
120,t,afgh,47,46987,120
```

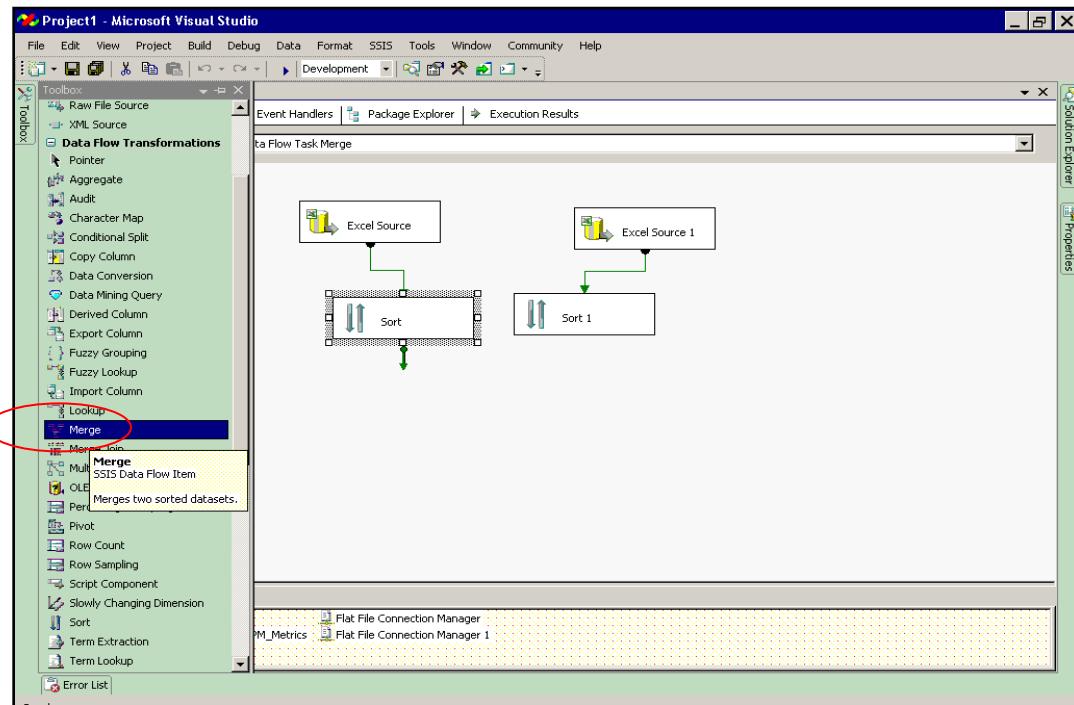
Lab 7- Merge Transformation

Objective	To merge data from two different sources into one target.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS project and Package

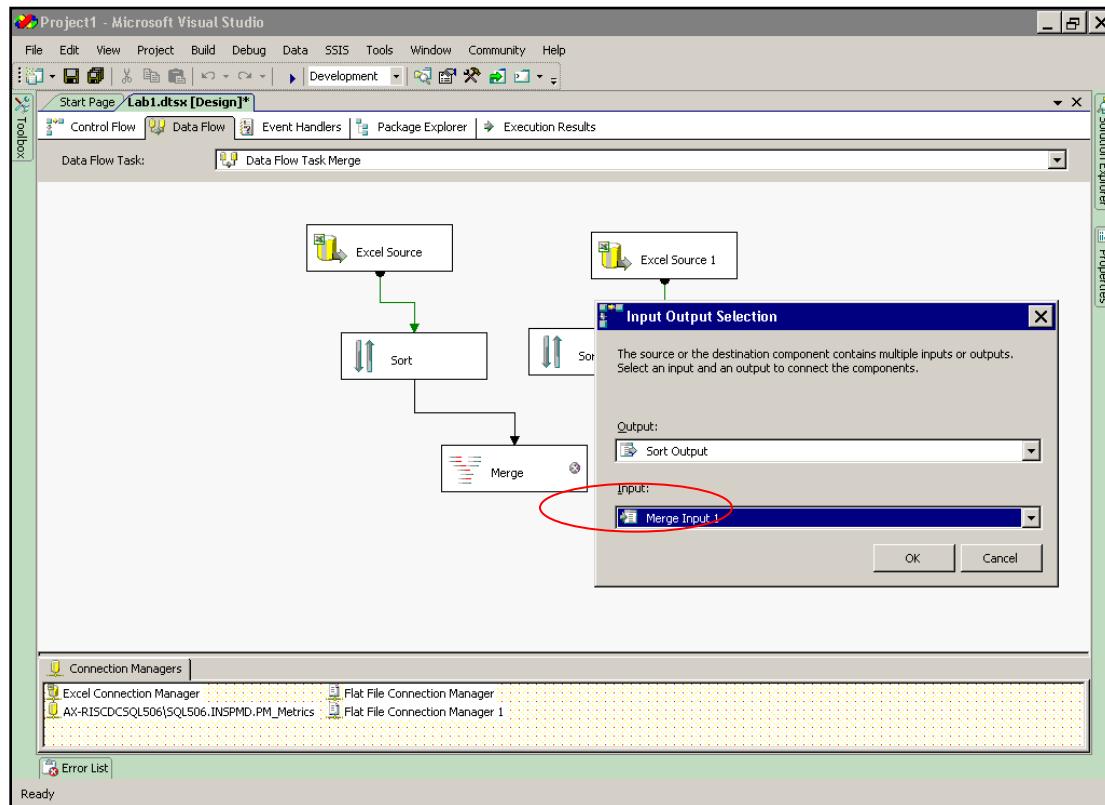
- 1)Select two excel file as source and edit the connection manager.
- 2)Select two sort transformations and edit the settings.



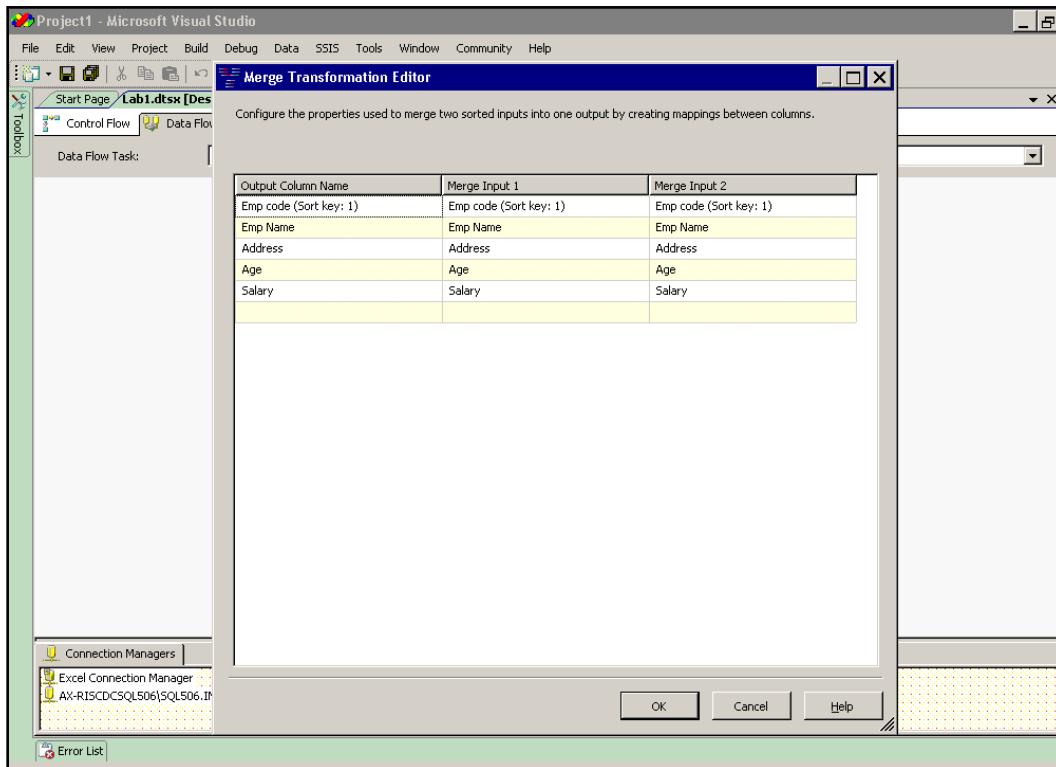
- 3)Sort both the source on the same column.
- 4)Then select merge transformation and join the sorted source to it.



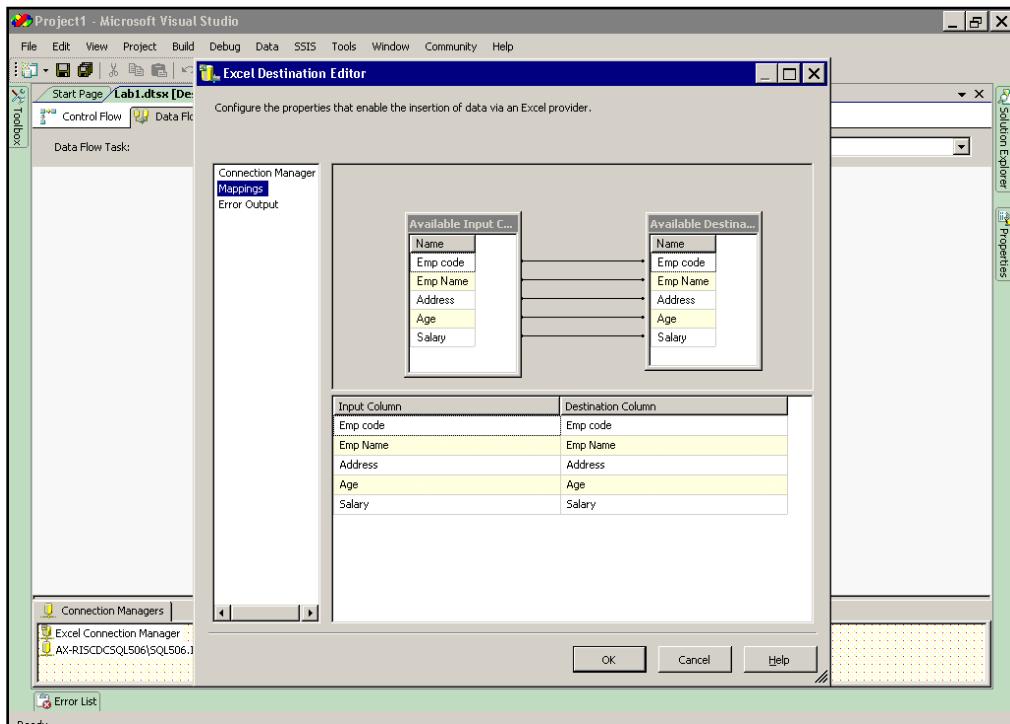
5) After joining the first SORT to the merge transformation, the window opens as:



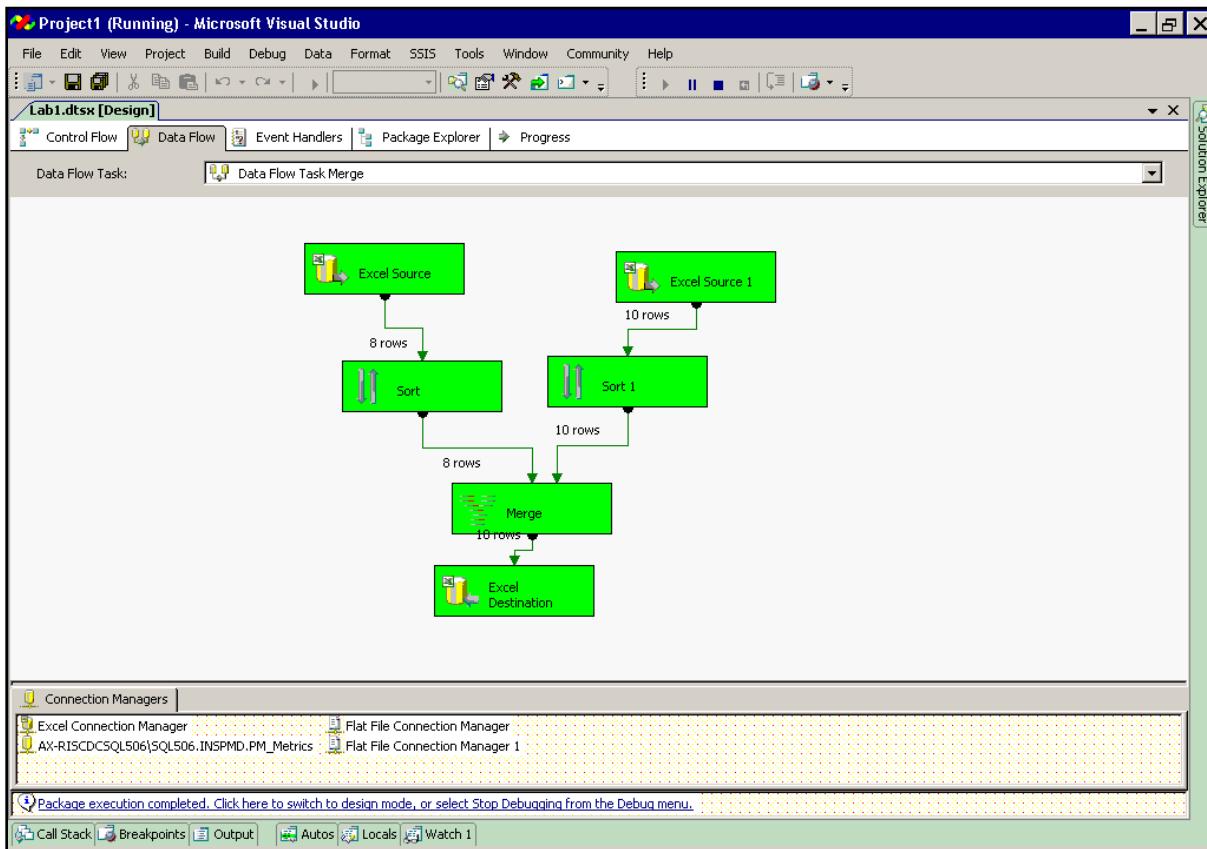
- 6)Select Sort Output and Merge Input 1
- 7)And then join the second sort to the Merge transformation.
- 8) Double click the merge transformation to view the editor.



- 9)Select a destination file (here we have taken excel as target)
 - Set the connection manager and map the columns in the Columns tab.



10) Save and execute the data flow task.



Data 1:

Emp code	Emp Name	Address	Age	Salary
112 l	vwx		29	31000
113 m	yz		30	32000
114 n	qwe		40	49875
101 a	klasfnx		23	20000
102 b	abc		43	56000
103 c	pqr		34	40000
104 d	asd		56	59000
105 e	abc		33	39000

Data 2:

Emp code	Emp Name	Address	Age	Salary
101 a	klasfnx		23	20000
102 b	abc		43	56000
103 c	pqr		34	40000

104 d	asd	56	59000
105 e	abc	33	39000
119 s	mnb	36	35647
120 t	afgh	47	46987
112 l	vwx	29	31000
113 m	yz	30	32000
114 n	qwe	40	49875

Output data:

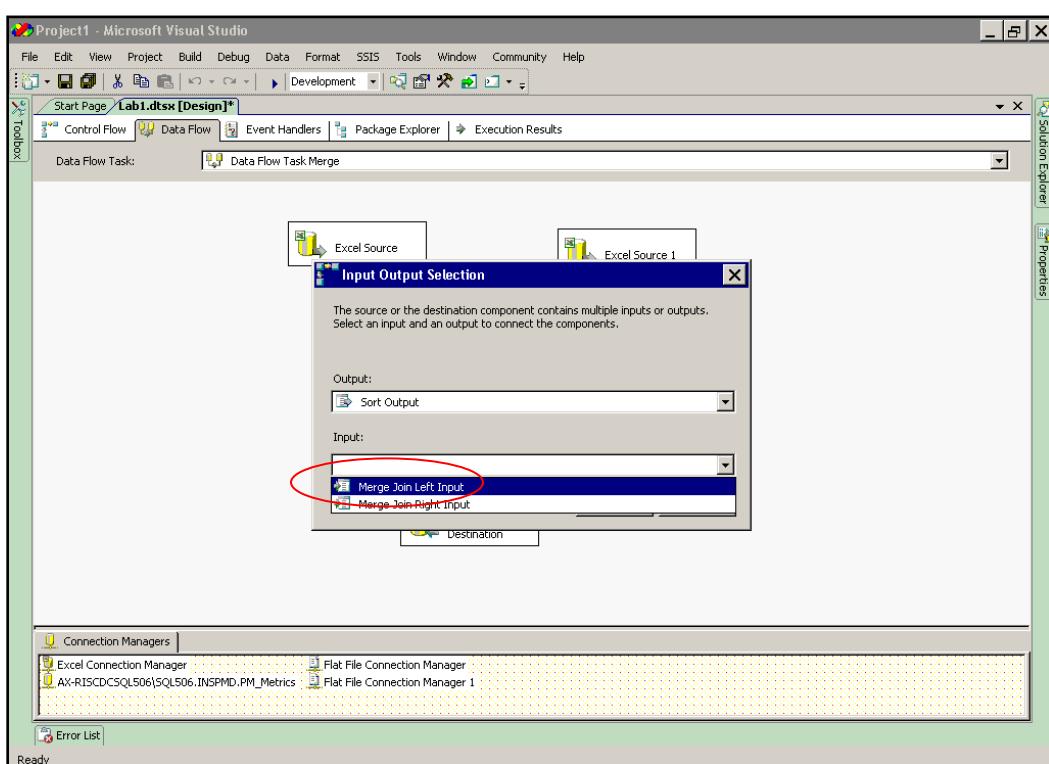
Emp code	Emp Name	Address	Age	Salary
101 a	klasfnx		23	20000
101 a	klasfnx		23	20000
102 b	abc		43	56000
102 b	abc		43	56000
103 c	pqr		34	40000
103 c	pqr		34	40000
104 d	asd		56	59000
104 d	asd		56	59000
105 e	abc		33	39000
105 e	abc		33	39000
112 l	vwx		29	31000
112 l	vwx		29	31000
113 m	yz		30	32000
113 m	yz		30	32000
114 n	qwe		40	49875
114 n	qwe		40	49875
119 s	mnb		36	35647
120 t	afgh		47	46987

Lab 8- Merge join Transformation

Objective	To merge data from two data sources and load into a target table using join condition.
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Existing SSIS project and Package

1) Repeat the steps from selecting the source till sorting of data and then drag and drop Merge Join Transformation.

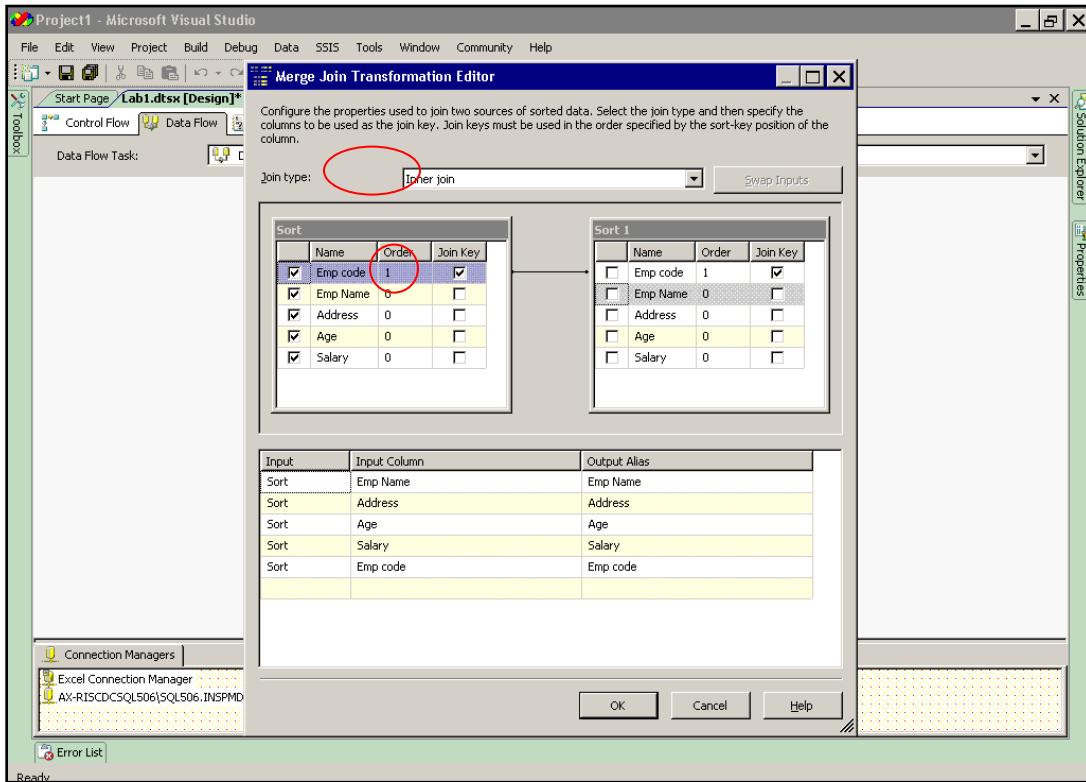
- ➔ Join the sort output to the transformation.
- ➔ The window opens as:



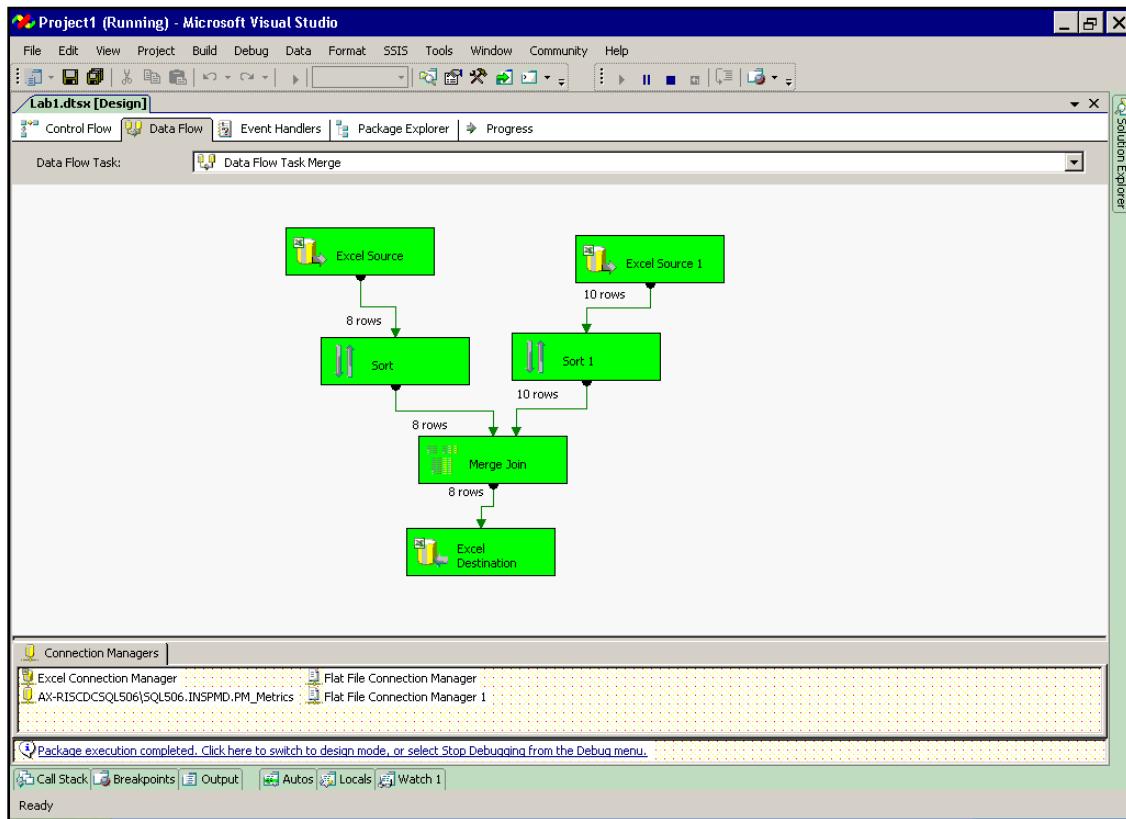
- ➔ Select the valid input type.
- ➔ Connect the second input to the transformation.

2) Double click merge join transformation to view the editor.

3) Select the required join type and the columns.



- 4) Connect the transformation to destination.
- 5) Edit the connection manager with valid connections and map the columns.
- 6) Save and execute the data flow.



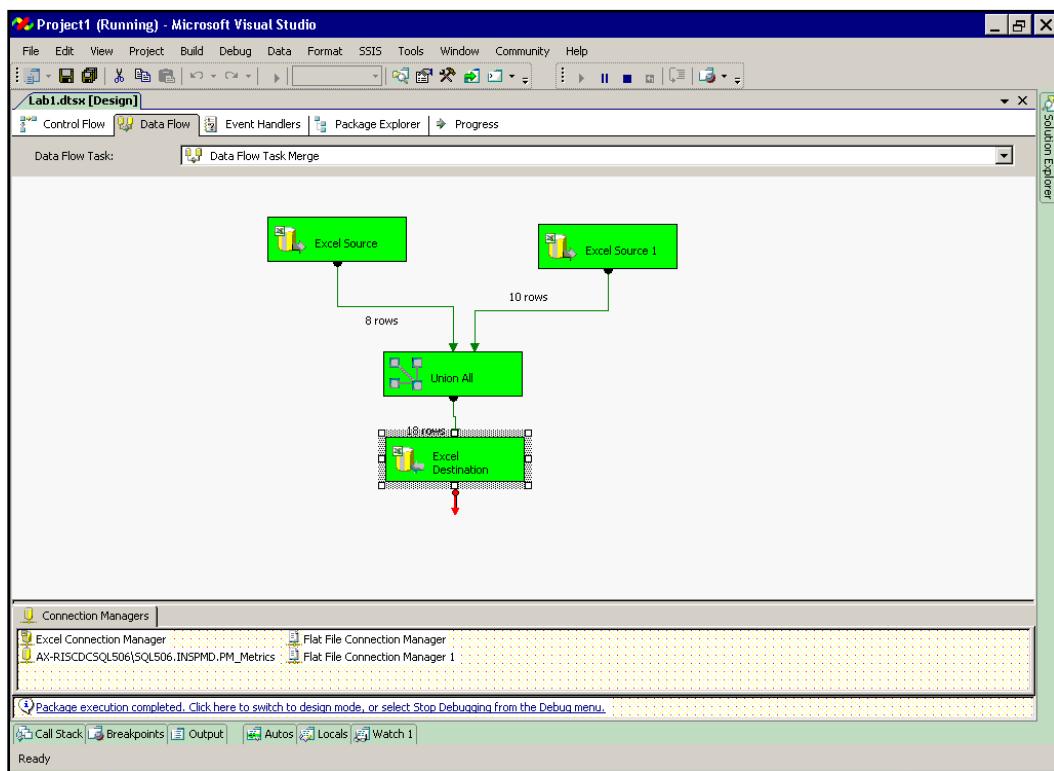
Output:

Emp Name	Address	Age	Salary	Emp code
a	klasfnx	23	20000	101
b	abc	43	56000	102
c	pqr	34	40000	103
d	asd	56	59000	104
e	abc	33	39000	105
f	vwx	29	31000	112
m	yz	30	32000	113
n	qwe	40	49875	114

Lab 9- Union All Transformation

Objective	To merge two data sources into one target table.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Existing SSIS project and Package

- 1)Select union all from the toolbox and connect it to the sources.
- 2)Select destination and join the union transformation to it. Edit the connection manager mapping.
- 3)Save and execute the data flow.



Output:

Emp code	Emp Name	Address	Age	Salary
112 l	vwx		29	31000
113 m	yz		30	32000
114 n	qwe		40	49875
101 a	klasfnx		23	20000
102 b	abc		43	56000
103 c	pqr		34	40000
104 d	asd		56	59000
105 e	abc		33	39000

101 a	klasfnx	23	20000
102 b	abc	43	56000
103 c	pqr	34	40000
104 d	asd	56	59000
105 e	abc	33	39000
119 s	mnb	36	35647
120 t	afgh	47	46987
112 l	vwx	29	31000
113 m	yz	30	32000
114 n	qwe	40	49875

Notes:

The Merge Join transformation provides an output that is generated by joining two sorted datasets using a FULL, LEFT, or INNER join.

The Merge transformation combines two sorted datasets into a single dataset. The rows from each dataset are inserted into the output based on values in their key columns.

The Merge transformation is similar to the Union All transformations. Use the Union All transformation instead of the Merge transformation in the following situations:

- The transformation inputs are not sorted.
- The combined output does not need to be sorted.
- The transformation has more than two inputs.

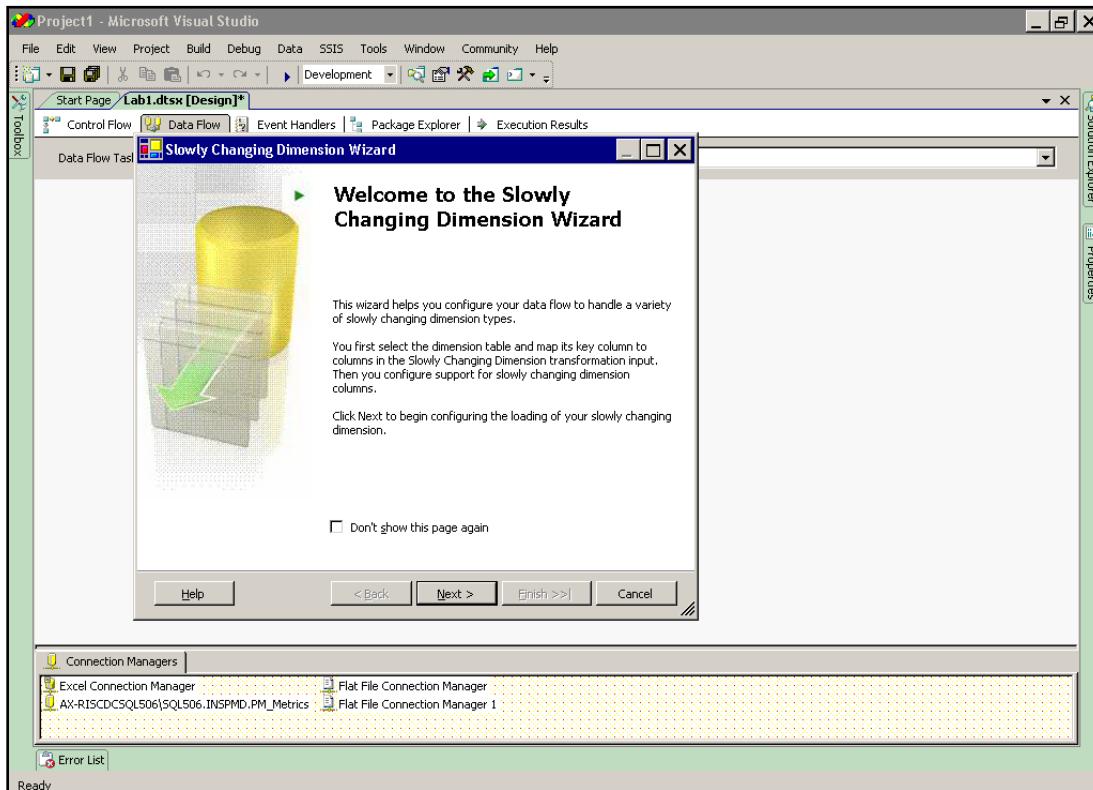
Lab 10-Slowly changing Dimension

Objective	To maintain the database in such a way that if any change occurs in the old data then the previous data is also shown along with the updated new data (Using Wizard)
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Existing SSIS project and Package

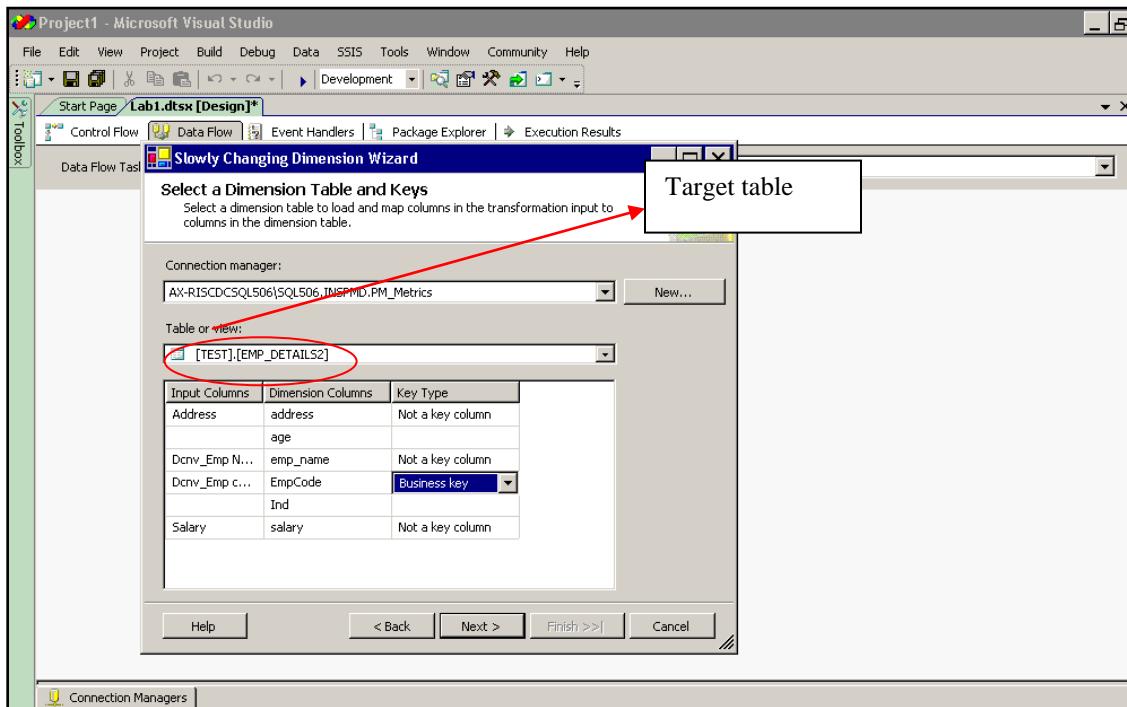
This transformation is used to maintain historical data.

- 1)Select a source (here we have used OLE DB)
- 2)Select Slowly Changing dimension transformation, Connect the source to it.

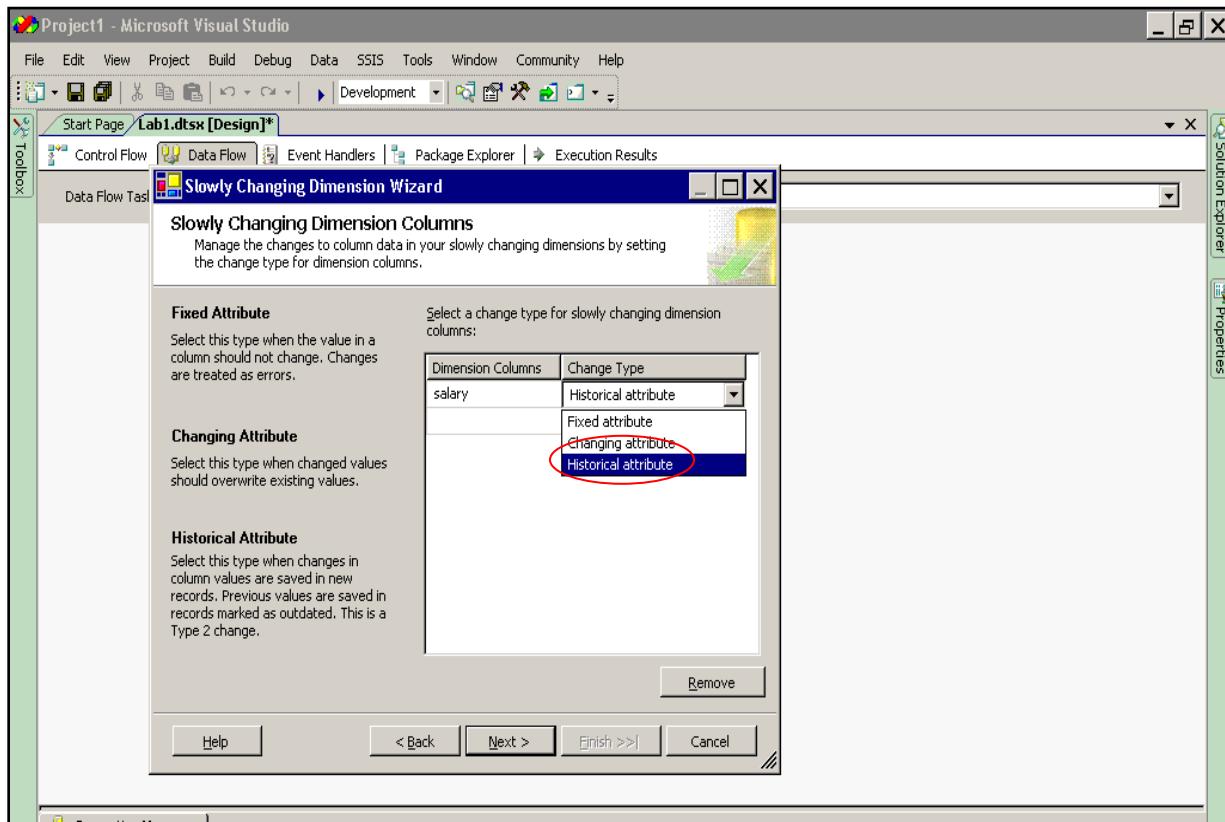
→ Double click the transformation to start the wizard.



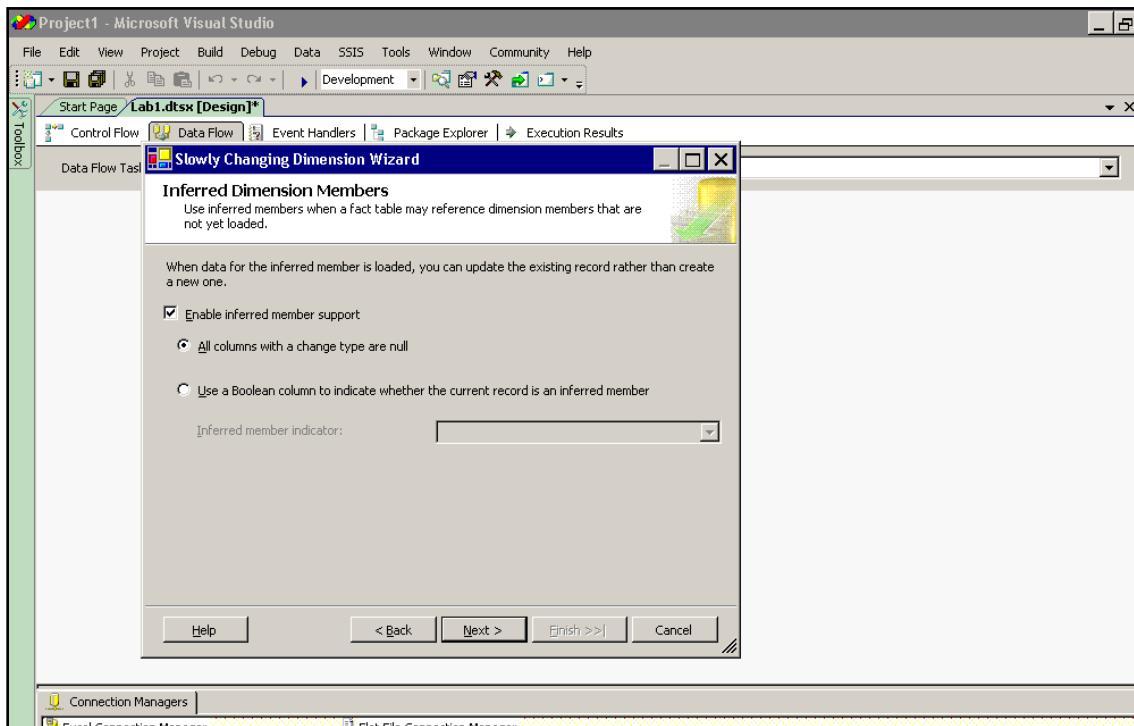
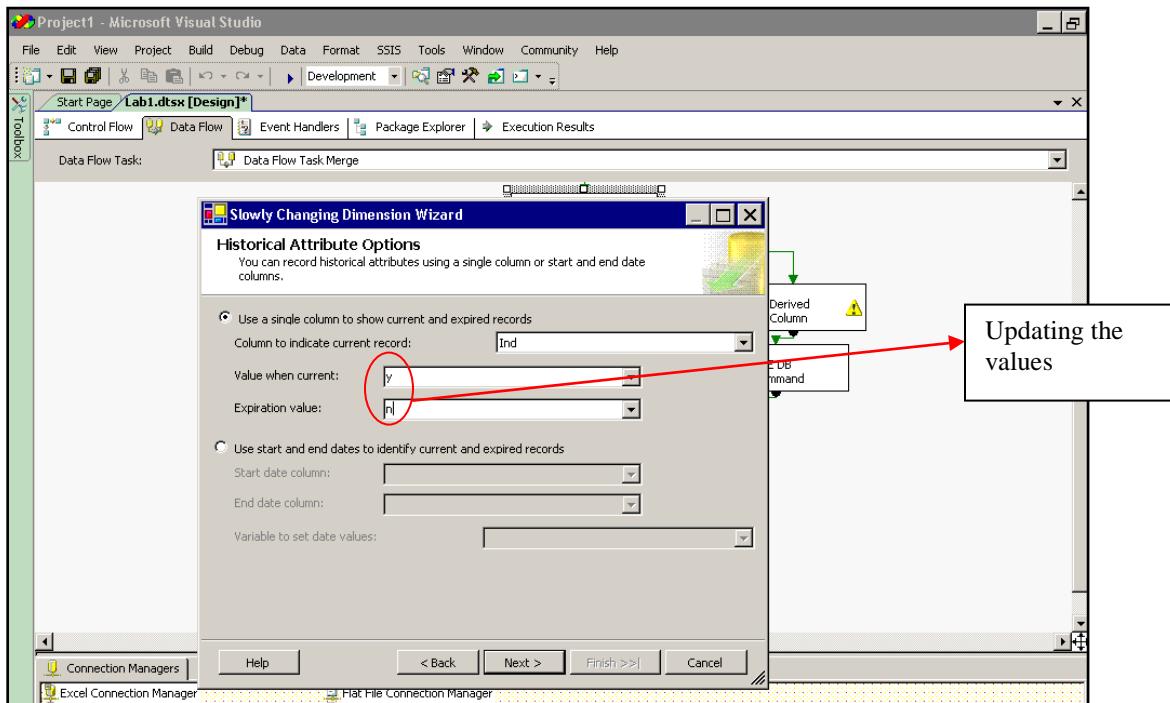
- Click on NEXT .
- Select a target table and then select input columns and use emp code as business key.



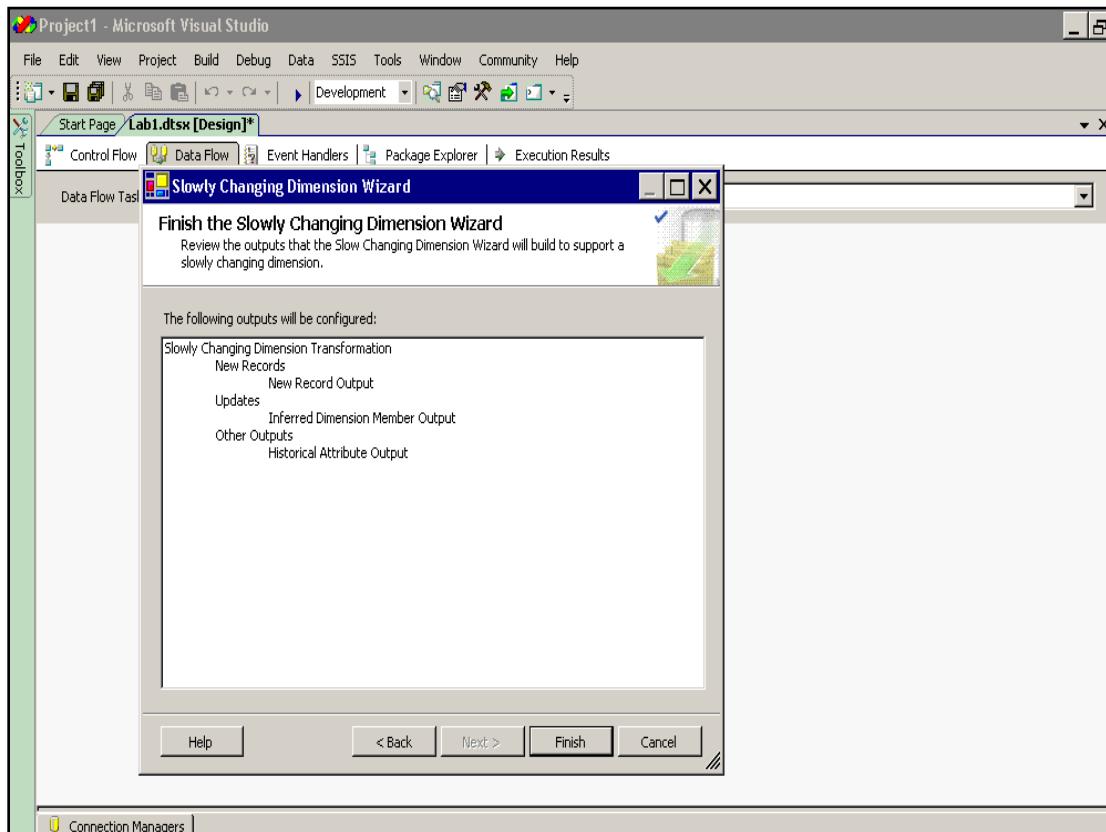
→ Now click on NEXT, select the column where the data changes and change type to **historical attribute**.



- Click NEXT.
- Use a column to differentiate new and old values.
Set the required fields and then click NEXT.

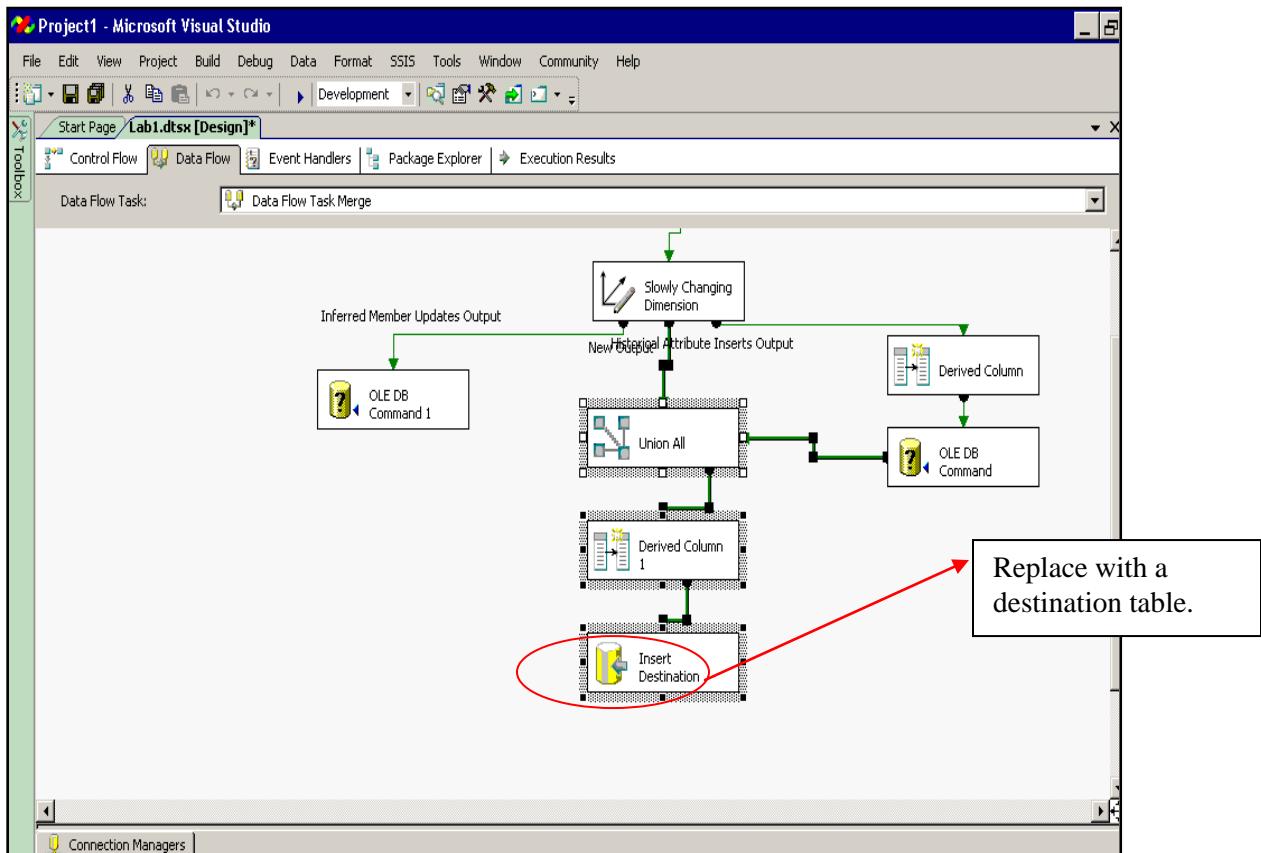


→ Click NEXT



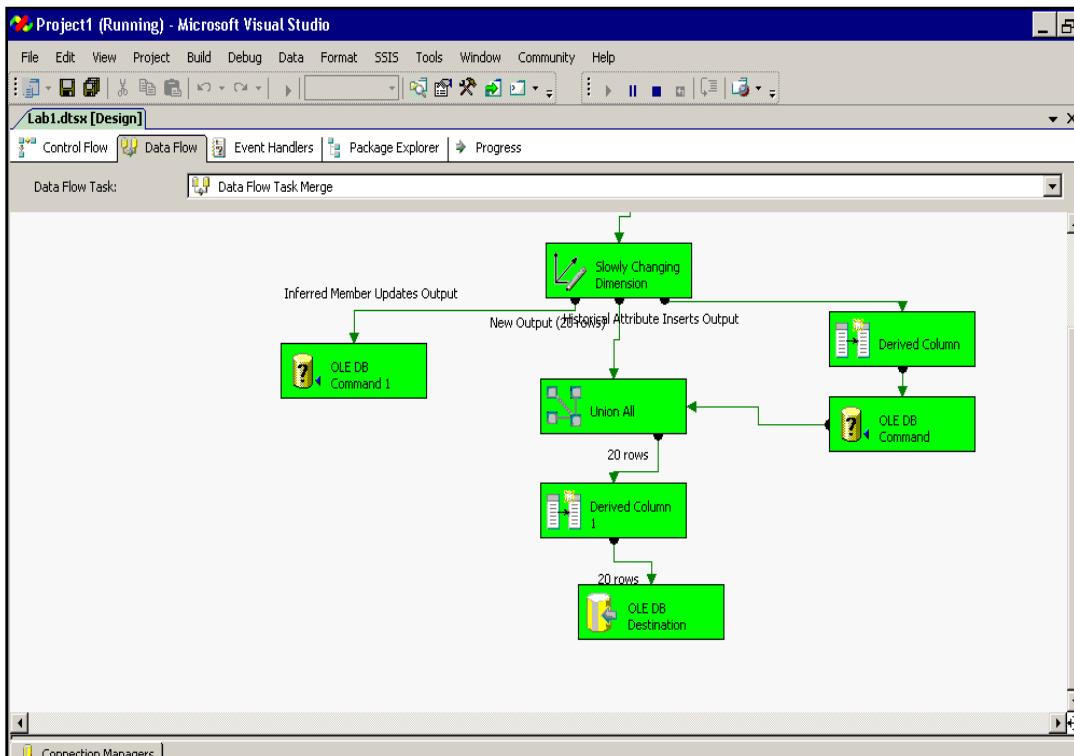
→ Click FINISH.

The output for the wizard will be like:



3) Update the destination with the target table.

4) Save and run the data flow.


Initial Load will show the data as:

Emp_code	emp_name	Address	Age	Salary	IND
101	a	klasfnx	23	20000	y
102	b	abc	43	56000	y
103	c	pqr	34	40000	y
104	d	asd	56	59000	y
105	e	abc	33	39000	y
106	f	perghjsd	27	36254	y
107	g	ghi	47	49000	y
108	h	jkI	55	70000	y
109	i	mno	28	30000	y
110	j	lop	38	36000	y
111	k	jshfkag	49	50000	y
112	l	vwx	29	31000	y
113	m	yz	30	32000	y
114	n	qwe	40	49875	y
115	o	rtyu	50	58000	y
116	p	fnfd	39	39785	y
117	q	sdfjfgosf	25	26000	y
118	r	jkg	26	27154	y
119	s	mnb	36	35647	y

120 t afgh 47 46987 y

Updating the rows:

```
update test.EMP_DETAILS
SET salary=23465 where EmpCode=101
update test.EMP_DETAILS
SET salary=50200 where EmpCode=107
update test.EMP_DETAILS
SET salary=26987 where EmpCode=117
```

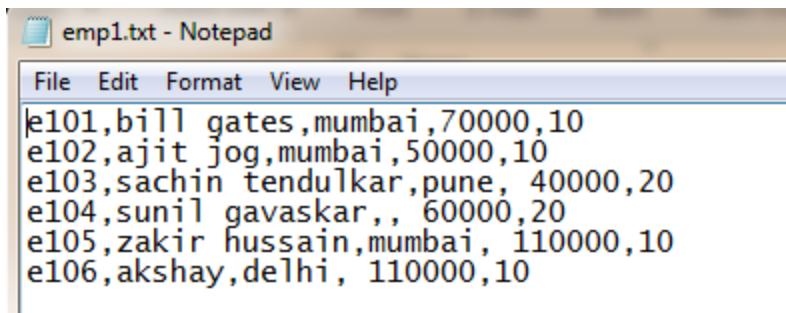
After changes in salary:

Emp_code	emp_name	Address	Age	Salary	IND
101 a	klasfnx		23	23564	n
102 b	abc		43	56000	y
103 c	pqr		34	40000	y
104 d	asd		56	59000	y
105 e	abc		33	39000	y
106 f	perghjsd		27	36254	y
107 g	ghi		47	50000	n
108 h	jkl		55	70000	y
109 i	mno		28	30000	y
110 j	lop		38	36000	y
111 k	jshfkag		49	50000	y
112 l	vwx		29	31000	y
113 m	yz		30	32000	y
114 n	qwe		40	49875	y
115 o	rtyu		50	58000	y
116 p	fnfd		39	39785	y
117 q	sdfjfgosf		25	27564	n
118 r	jkg		26	27154	y
119 s	mnb		36	35647	y
120 t	afgh		47	46987	y
101 a	klasfnx		23	23465	y
107 g	ghi		47	50200	y
117 q	sdfjfgosf		25	26987	y

Lab 11- Extracting, Transforming and Loading Data

Objective	To learn <ol style="list-style-type: none"> 1. How to extract data from flat file 2. How to Transform Data <ol style="list-style-type: none"> a. Add new Derived Column b. Change Data Types c. Perform a Data Lookup d. Conditionally route and load data into relational table 3. Load data into relational table
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Existing SSIS project and connection to database

- 1)Create a separate folder “Demo2” for placing input data feed files
2)Place a emp1.txt data file in this folder with following data



```
emp1.txt - Notepad
File Edit Format View Help
e101,bill gates,mumbai,70000,10
e102,ajit jog,mumbai,50000,10
e103,sachin tendulkar,pune, 40000,20
e104,sunil gavaskar,, 60000,20
e105,zakir hussain,mumbai, 110000,10
e106,akshay,delhi, 110000,10
```

IN SSMS:

- 3)Create “mumbaiemp” table in sql server database with following structure

```
CREATE TABLE [dbo].[mumbaiemp](
[empno] [varchar](7) PRIMARY KEY,
[empname] [varchar](20) NULL,
[city] [varchar](20) NULL,
[sal] [money] NULL,
[da] [money] NULL,
[hra] [money] NULL,
[deptno] [int] NULL,
[dname] varchar(20)
)
```

- 4)Create 2 more tables “puneemp” and “otheremp” with exact same structure as above.
5)Create dept table as below:

```
createtable dept
(
deptno int primary key,
deptname varchar(20)
)
```

- 6)Add the following records to the dept table:

	deptno	deptname
1	10	finance
2	20	sales
3	30	accounts
4	40	purchase
5	50	IT

IN SSIS:

- 1)In SSDT ,create New SSIS Package
 - Rename it as “FF_To_DB_With_Transformation.dtsx”
- 2)In the package designer, goto data flow tab and click the link in the middle to create a new data flow task.
- 3)Same as was done in Lab 1, drag a flat file source task from toolbox onto designer surface and make it point to emp1.txt file placed in Demo2 folder.
- 4)Next, drag derived column task from transformation section of toolbox
 - Connect the flat file task green connector to this task
 - Double click this task and configure the additional derived columns as shown below:

Derived Column Name	Derived Column	Expression	Data Type	L
FF_Col_DA	<add as new column>	FF_Col_EmpSal * 0.20	numeric [DT_NUMERIC]	
FF_Col_HRA	<add as new column>	FF_Col_EmpSal * 0.25	numeric [DT_NUMERIC]	

- 5)Drag a Data Conversion task and connect the Derived Column task green connector to this task.
 - Double click it and change the data type of the following to columns from DT_Numeric to DT_CY as shown below:

Input Column	Output Alias	Data Type	Length
FF_Col_DA	FF_Col_DA	currency [DT_CY]	
FF_Col_HRA	FF_Col_HRA	currency [DT_CY]	

- 6)Drag a lookup transform task and connect the data conversion task to it.
 - Double click lookup task, go to connection tab, create a new connection manager which will point to the sql server database which has “Dept” table created earlier in this lab.

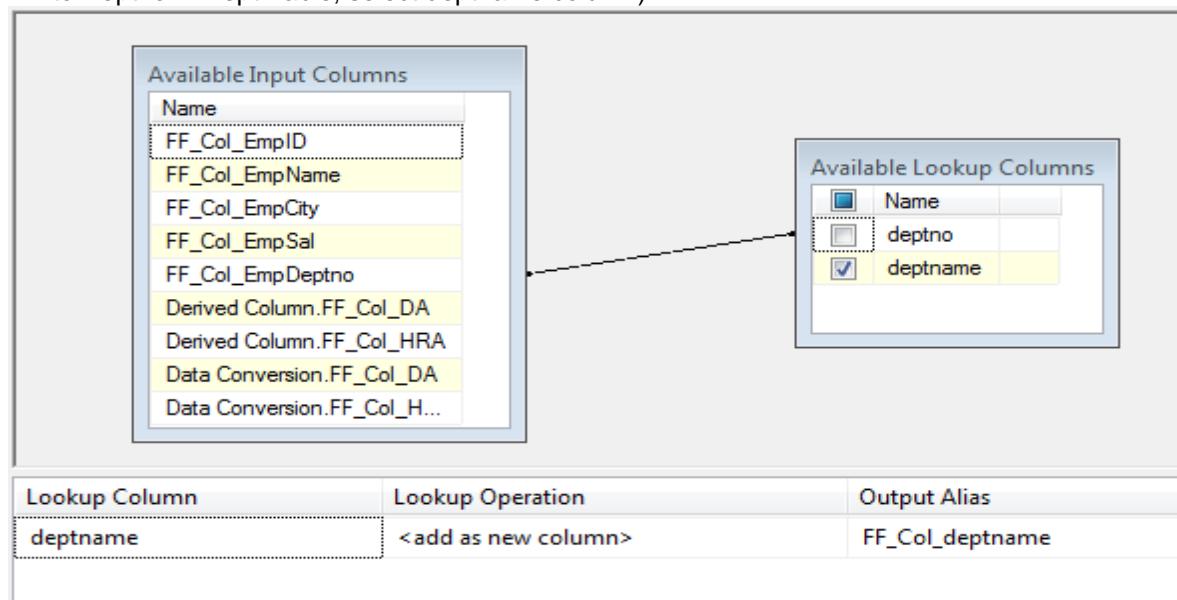
OLE DB connection manager:

New...

Use a table or a view:

New...

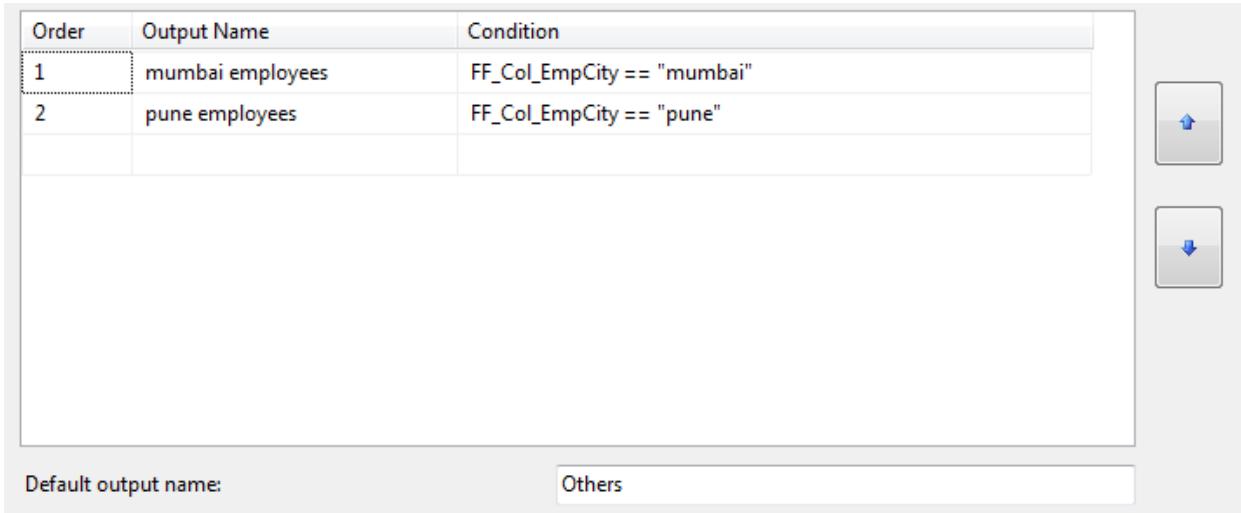
- Goto columns tab, and configure this tab as shown below. (connect FF_Col_EmpDeptno to Deptno in Dept Table, select deptname column)



Lookup Column	Lookup Operation	Output Alias
deptname	<add as new column>	FF_Col_deptname

7) Drag a conditional split task onto surface rename it as “Split Employee records based on city”, connect the green connector of lookup task to this task a dialog will popup, from drop down select “lookup match output”.

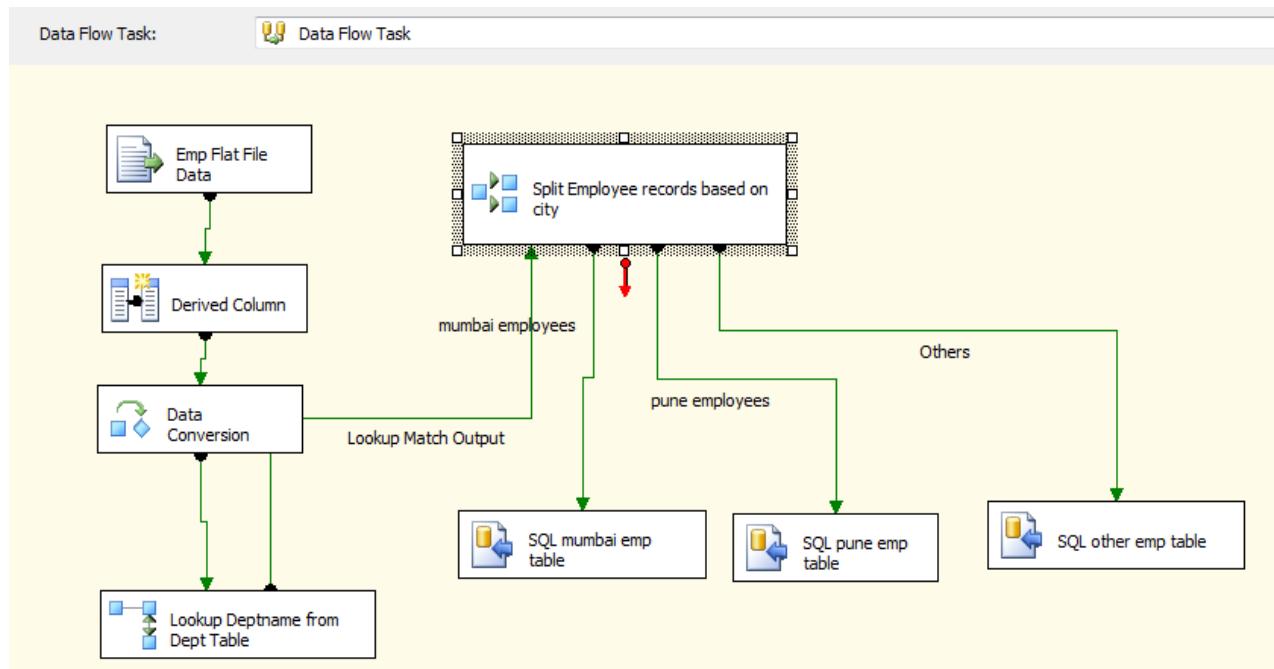
- Double click conditional split task and configure as below: (there are 2 named outputs which will send only those rows that satisfy the specified condition, and there is default output as well called “others”)



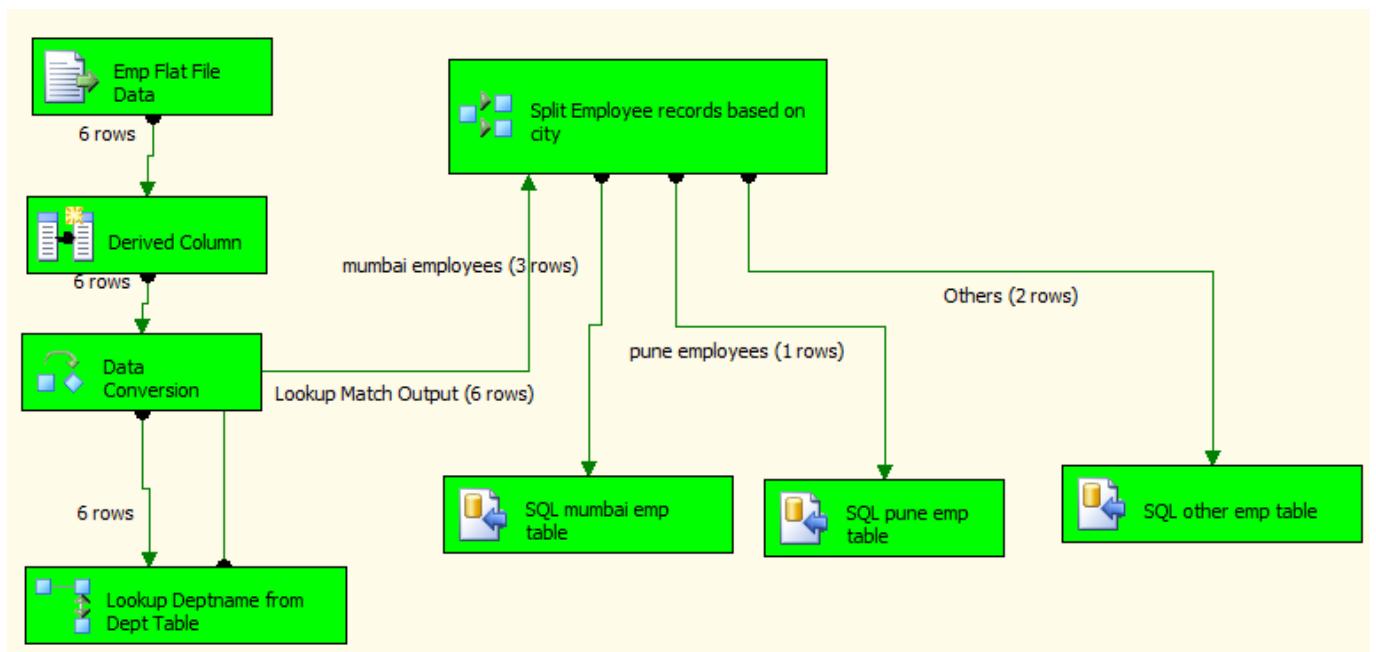
8) Drag 3 SQL Server Destinations tasks from data flow destinations section of toolbox. Rename them as “SQL Mumbai emp table”, SQL Pune Emp Table, SQL Other emp table.

9) Connect conditional split task to the first “SQL Mumbai emp table” sql server destination task. A dialog box will popup select the output name to connect as “mumbai employees”. Similarly connect the remaining outputs to the corresponding tasks.

10) Configure each of the these tasks by double clicking them so that they point to appropriate tables in the sql server. The complete SSIS package is shown below:

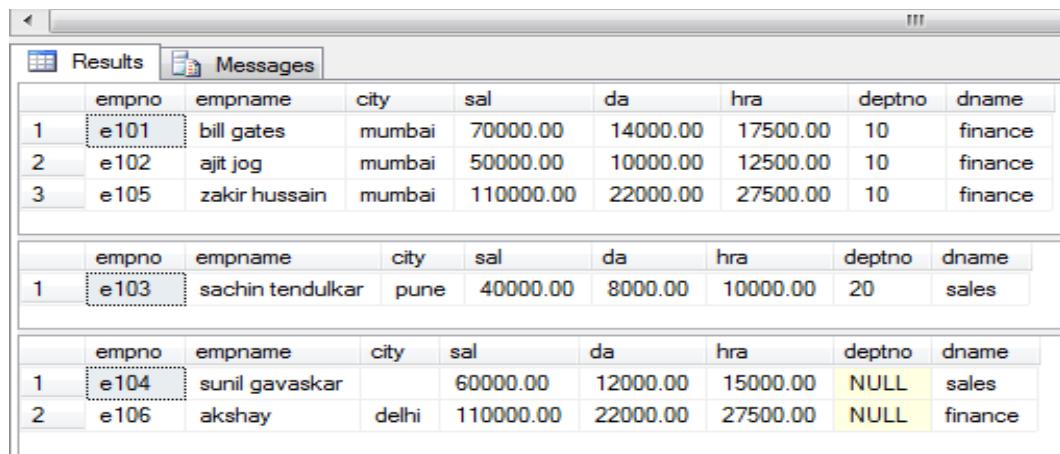


11) Run the package, when the entire package executes successfully it will look as shown below:



12)Stop the Package. 

13)Connect to sql server using SSMS and check the table data



The screenshot shows the SSMS Results window displaying data from three different tables:

SQL mumbai emp table								
empno	empname	city	sal	da	hra	deptno	dname	
1	e101	bill gates	mumbai	70000.00	14000.00	17500.00	10	finance
2	e102	ajit jog	mumbai	50000.00	10000.00	12500.00	10	finance
3	e105	zakir hussain	mumbai	110000.00	22000.00	27500.00	10	finance

SQL pune emp table								
empno	empname	city	sal	da	hra	deptno	dname	
1	e103	sachin tendulkar	pune	40000.00	8000.00	10000.00	20	sales

SQL other emp table								
empno	empname	city	sal	da	hra	deptno	dname	
1	e104	sunil gavaskar		60000.00	12000.00	15000.00	NULL	sales
2	e106	akshay	delhi	110000.00	22000.00	27500.00	NULL	finance

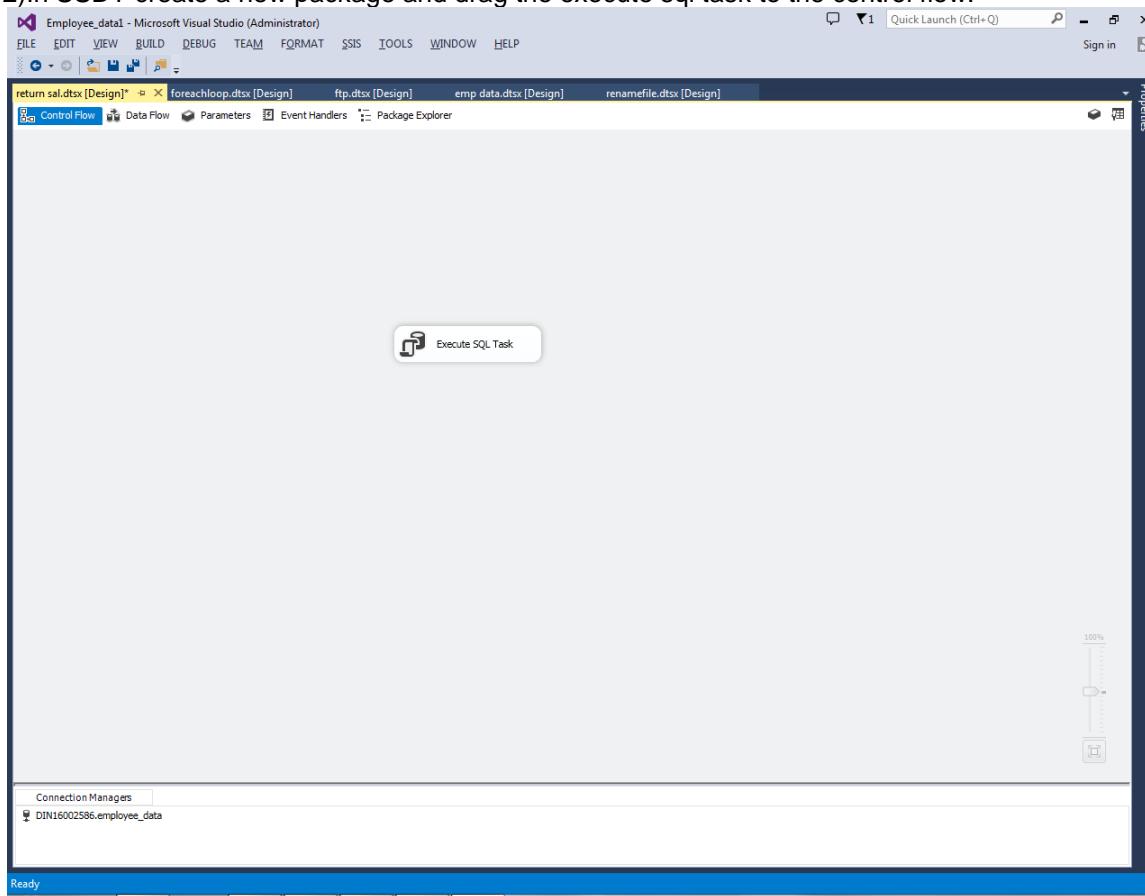
Lab 12- Execute SQL Task& Script task

Objective	Execute a stored procedure from SSIS to get employee salary and display output in message box
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Existing SSIS project and connection to database

1)As per below create a stored procedure to get employee salary as outcome when passing salary name as input parameter.

```
createprocedure [dbo].[emp_name9](
@emp_name varchar(20),
@salary intoutput
)
as
begin
select @salary=[salary] from [dbo].[empdata] where [emp_name]=@emp_name
end
```

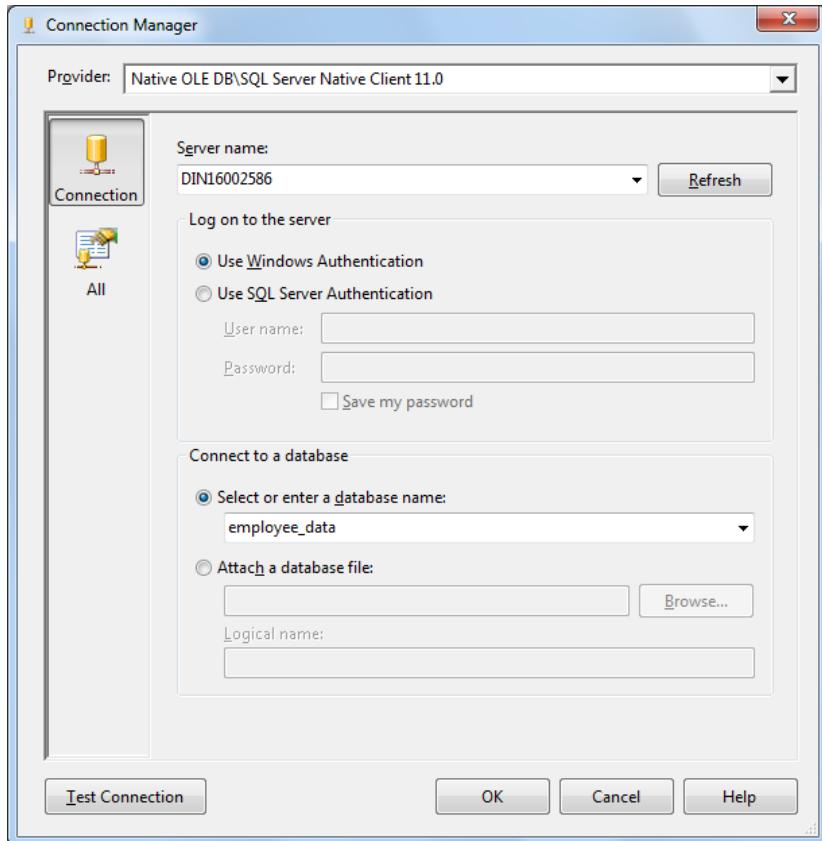
2)In SSDT create a new package and drag the execute sql task to the control flow.



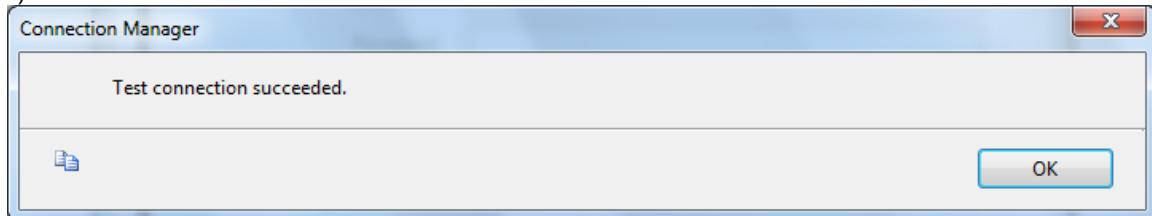
3)Create a variable as per below. You can create variables by right clicking on the control flow or from the SSIS menu above the control flow.

Name	Scope	Data type	Value	Expression
salary	return sal	Int32	0	...

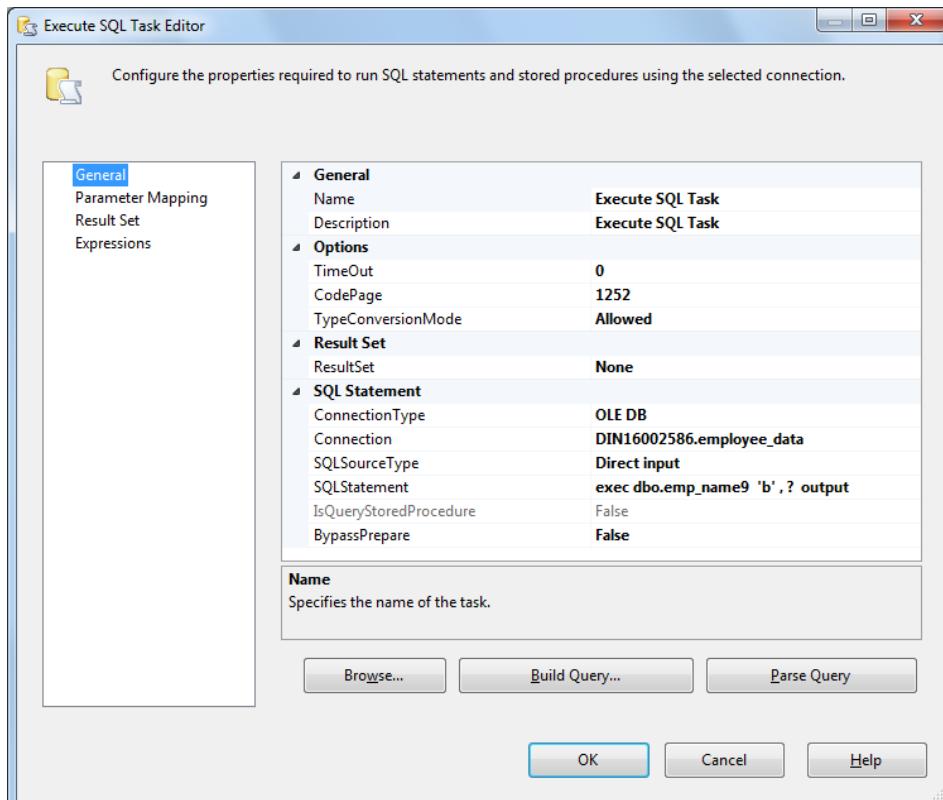
4)Create a new connection string as per below. If connection string already exists no need to create new connection string, you can use the same connection string.



- 5)Choose the provider as SQL Server Native Client 11.0.
6)Choose your server name and database which holds your employee_data table.
7)Click the Test Connection tab and test the connection.



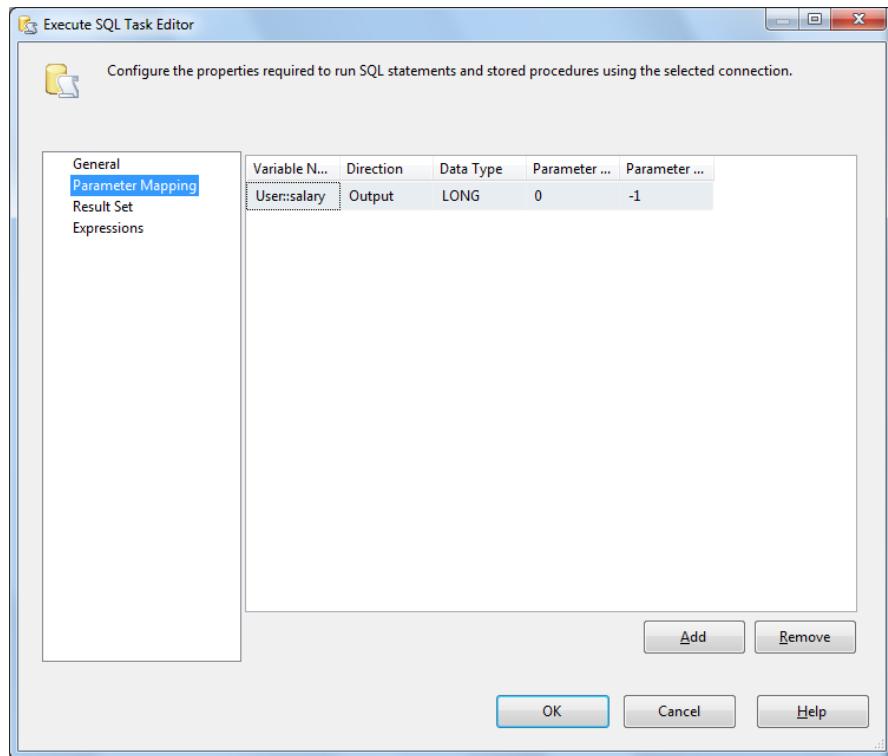
- 8)Edit the Execute SQL task and choose the connection string which you have created and SQL Source type as Direct Input as per below.



5)Provide the execute statement in SQLStatement field as per below.

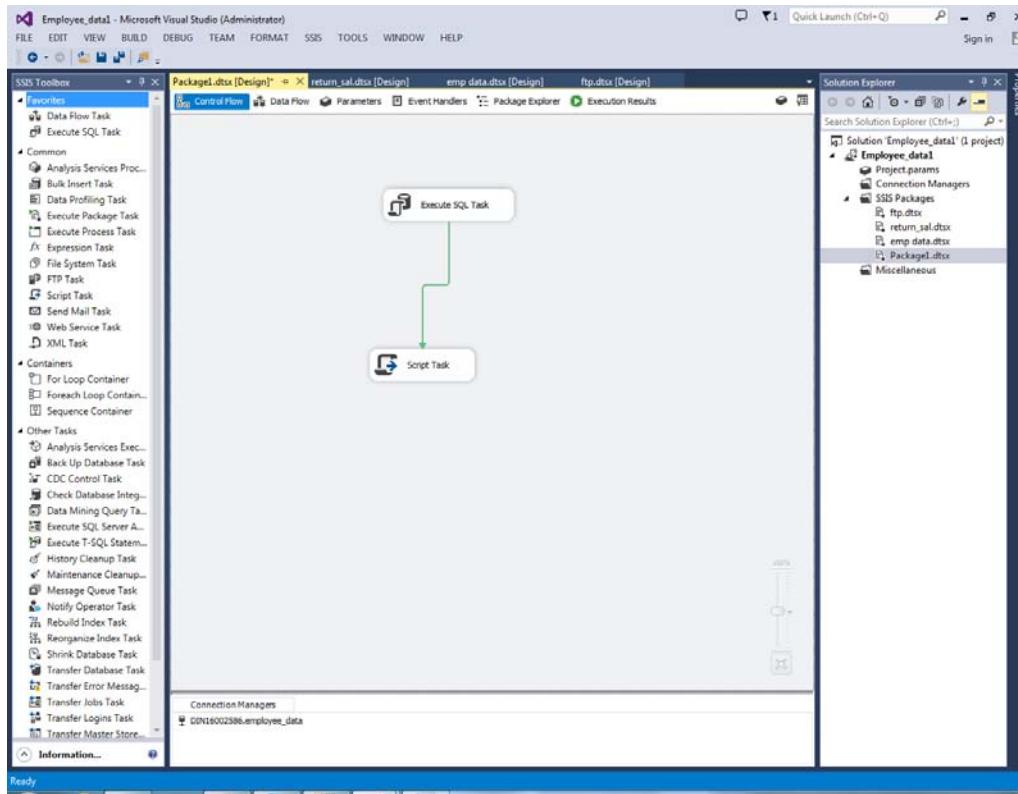
SQL Statement: exec dbo.emp_name9 'p', ? output

6)In parameter mapping choose the parameter you have created and give the direction as output and parameter name as 0.

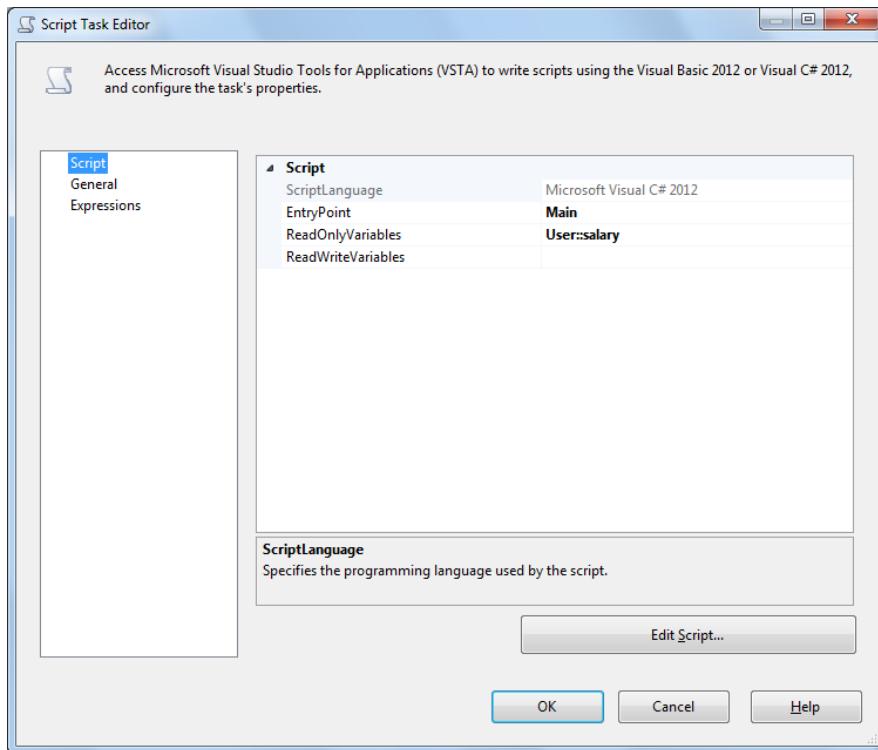


7)Now will use Script task to display salary of the Employee that we have stored in output variable.

8)Drag a Script task to the control flow and connect the green arrow from execute sql task to the script task.



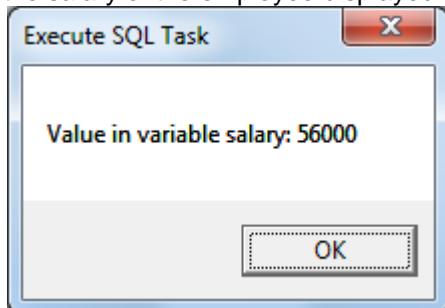
9)Now edit the script task and in the read only variable choose the output parameter **User::salary**



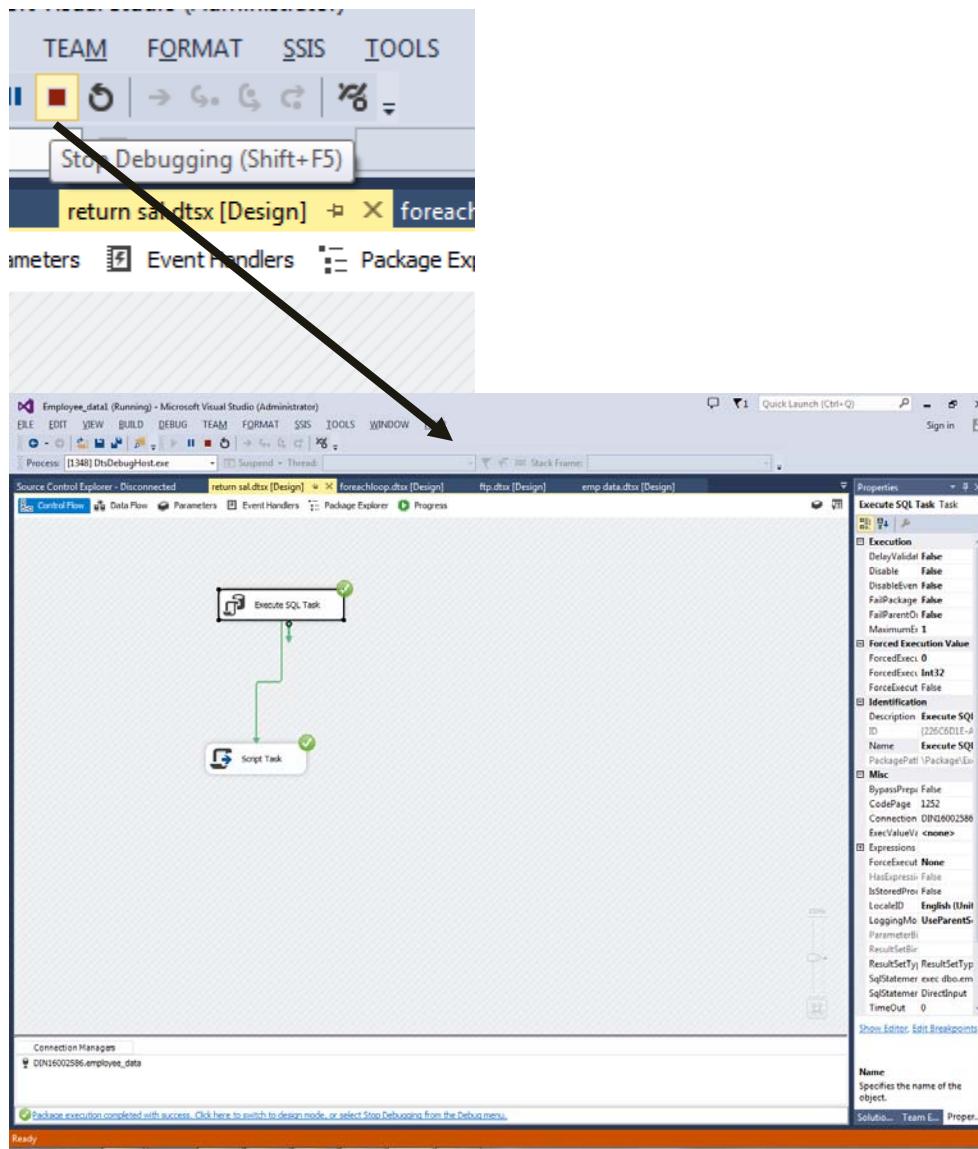
10)In Edit Script and add the below code in main function to display the salary of the employee.

```
MessageBox.Show(String.Format("Value in variable salary: {0}",  
Dts.Variables["User::salary"].Value), "Execute SQL Task");
```

11)Now execute the package by right clicking the project name in the solution explorer and you can see the salary of the employee displayed in the message box.



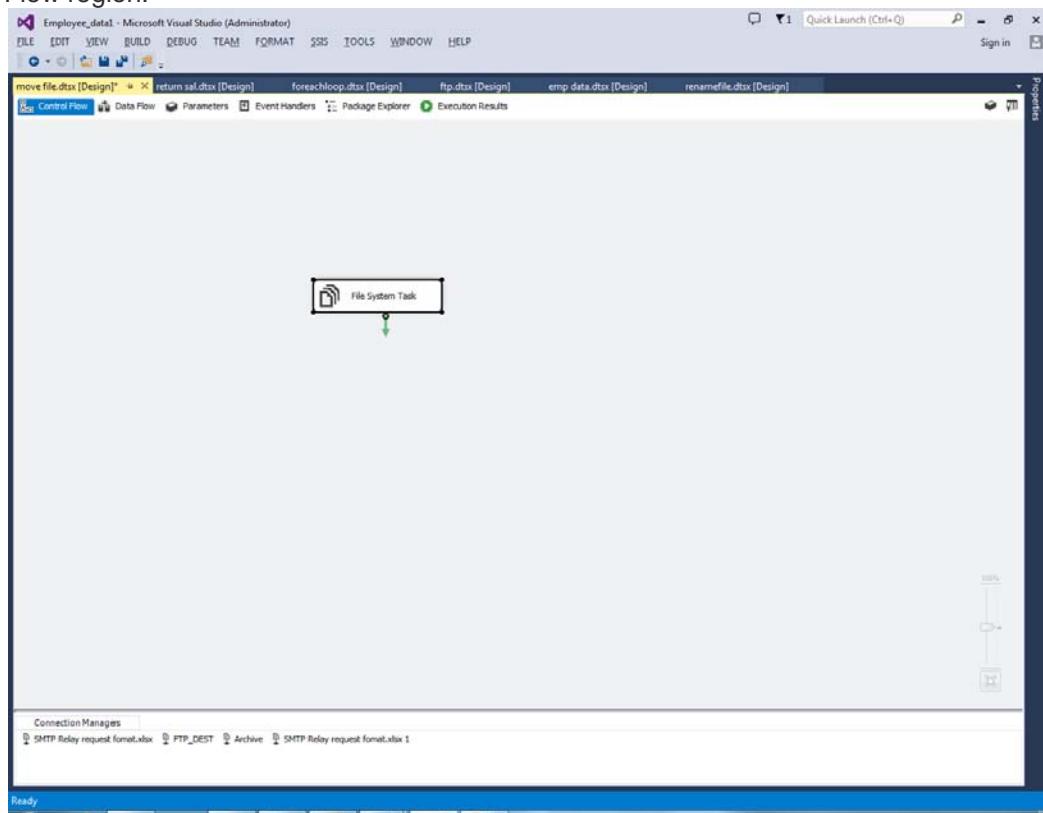
12)After executing stop debugging to return to the design mode.



Lab 13-File System Task

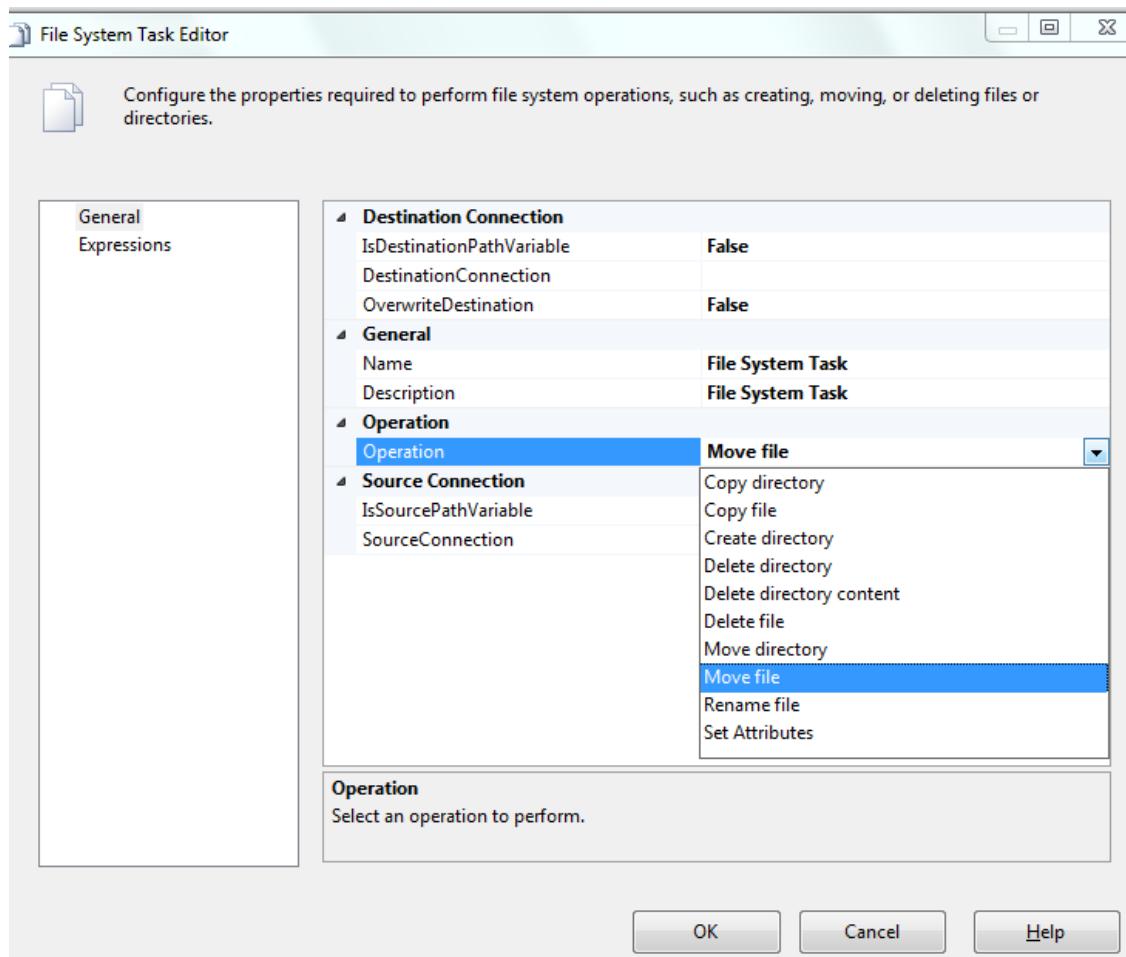
Objective	To learn 1.How to move file from one location to another location.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Create package

1)In order to move file using file system task , drag and drop the File System Task into the Control Flow region.

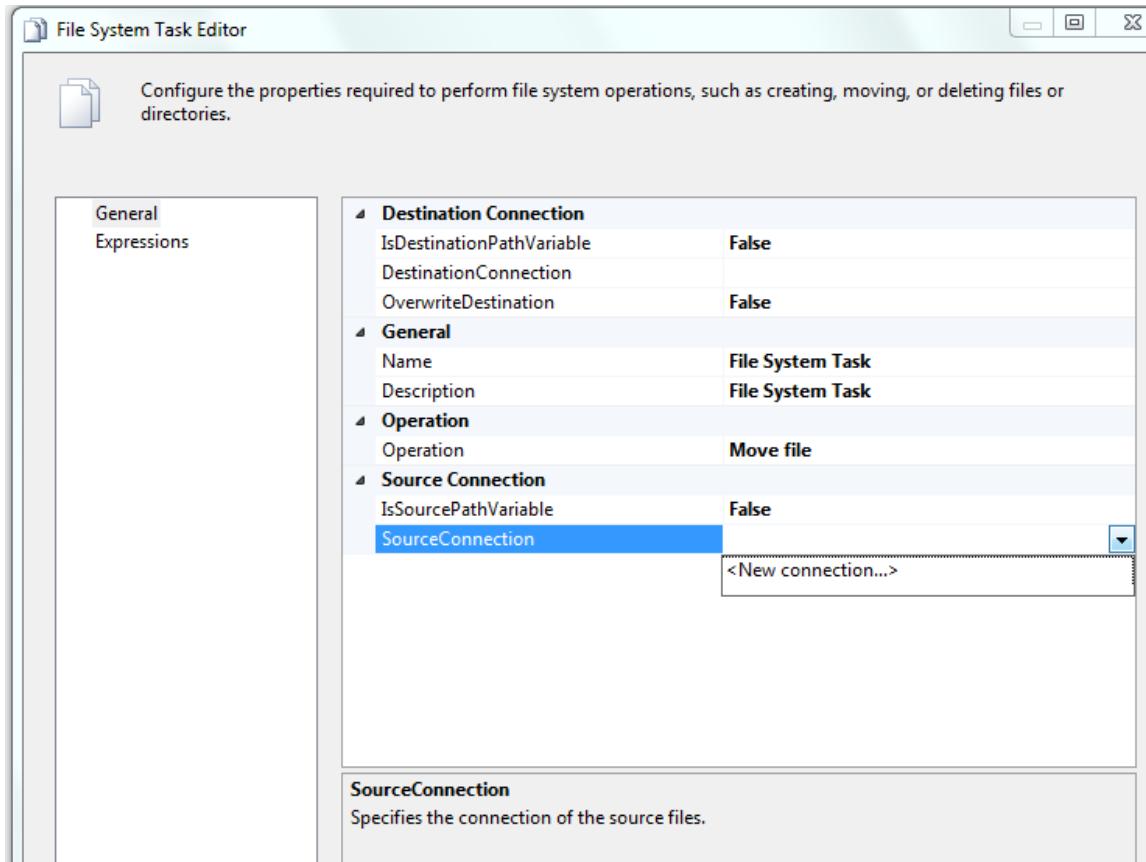


2) Double click on it will open the File System Task Editor to configure it.

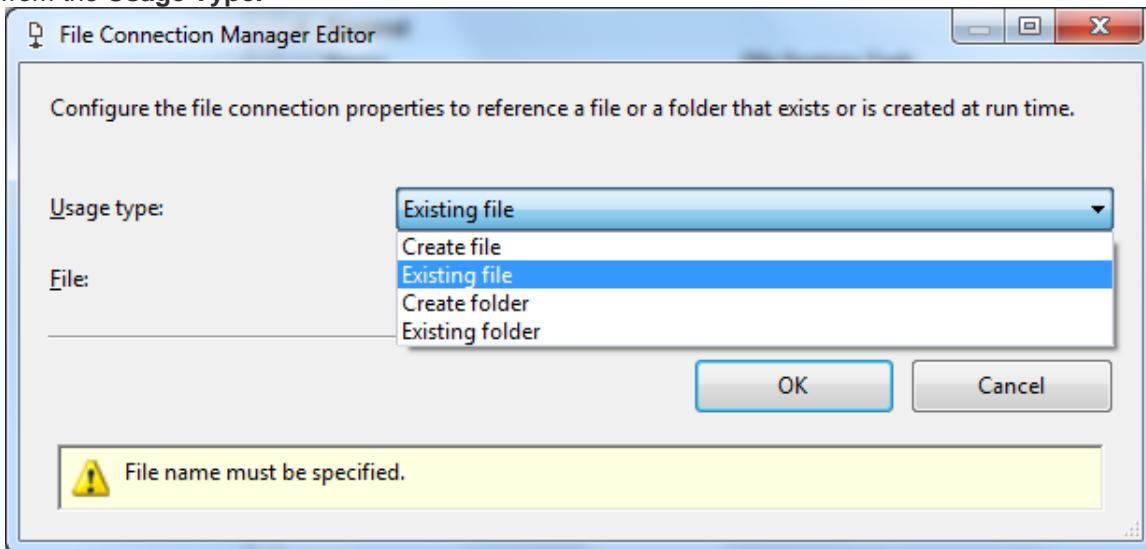
3)In this example, We are Moving single file so, Please change the **operation** property to **Move File** as shown in the below screenshot.



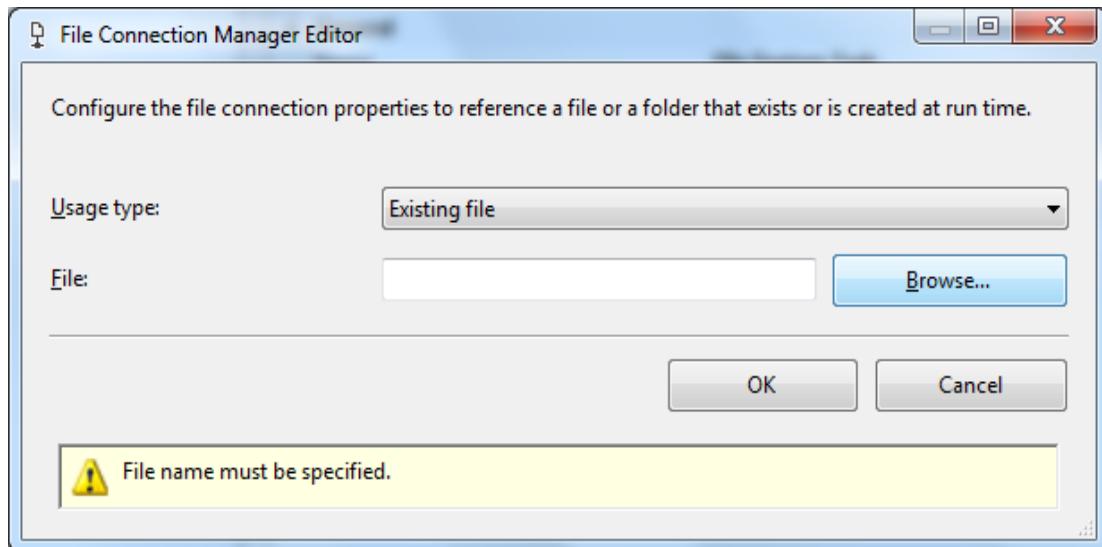
- 4) Let us configure the Source Connection by selecting the **Source Connection** property.
- 5) If you already created the File Connection Manager before then select the created one or If you stored the Source Connection in the Variable then, please change the **IsSourcePathVariable** property to TRUE and select the Variable Name.
- 6) Here, We haven't created any connection Manager before so, We are selecting <New Connection...>



7)Once you click on the <New Connection..> option, File Connection Manager Editor will be opened to configure it. In this example we are Moving existing file so we are selecting Existing File option from the **Usage Type**.

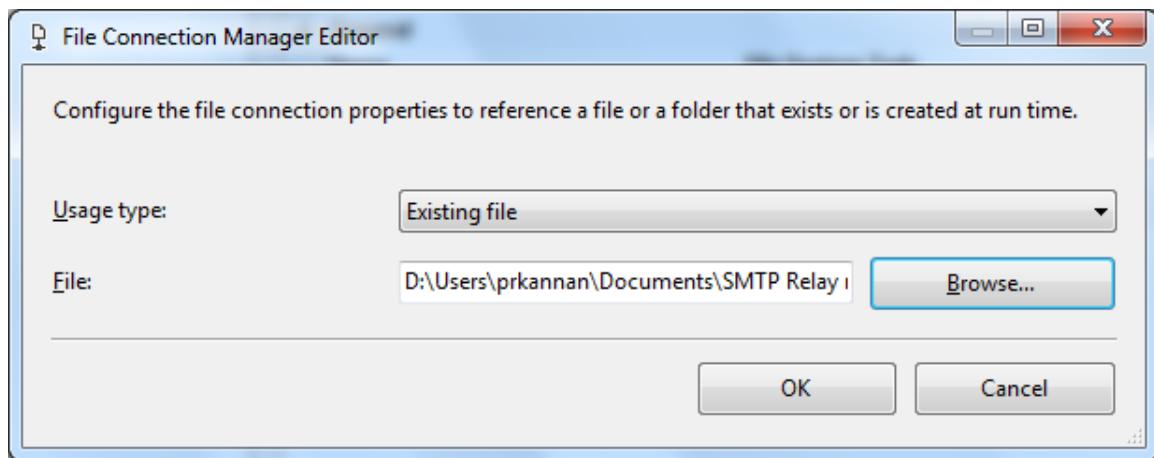


8)Click on the Browse button to select the Existing File from the file system.



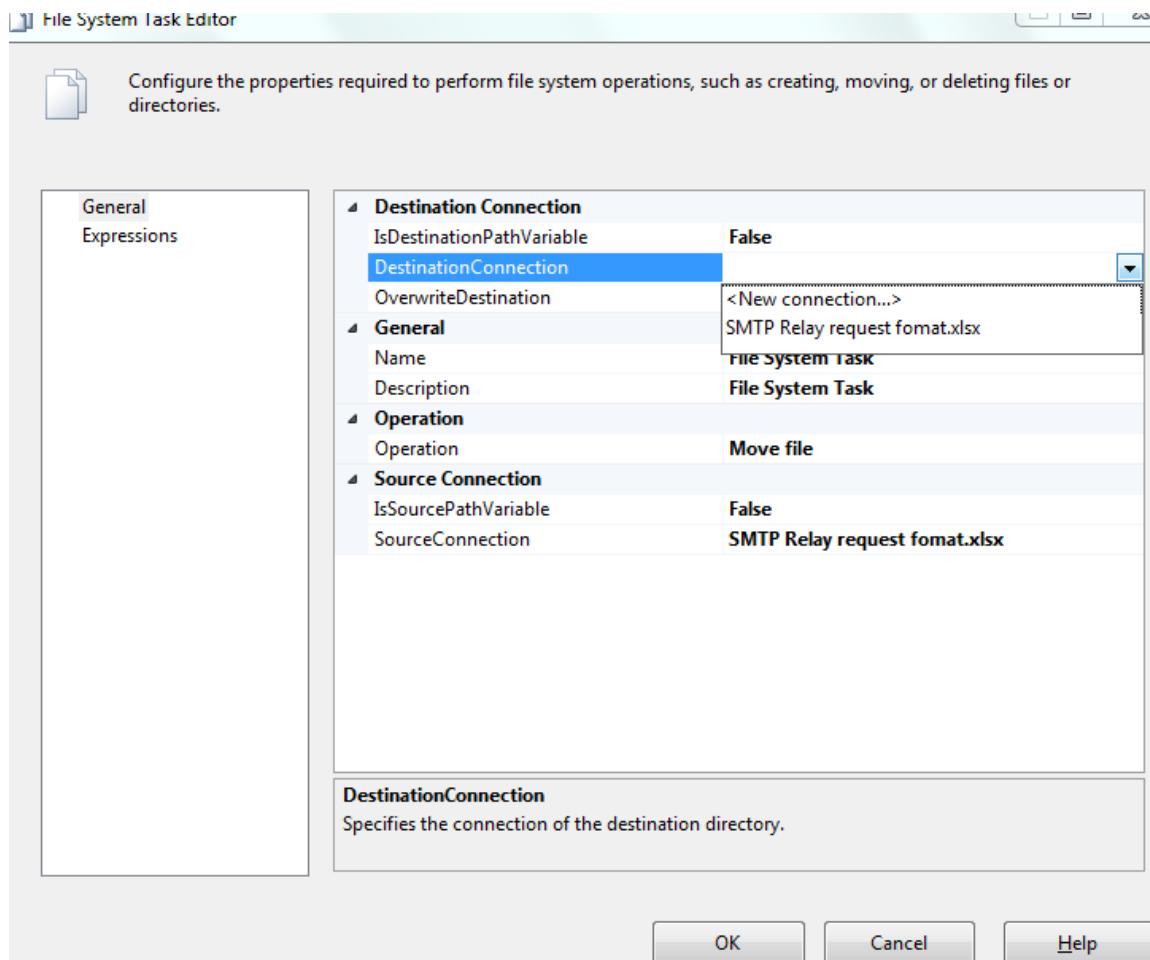
9)

From the above screenshot you can observe that, We selected the SMTP Relay request format.xlsx file inside the File System Task Folder.

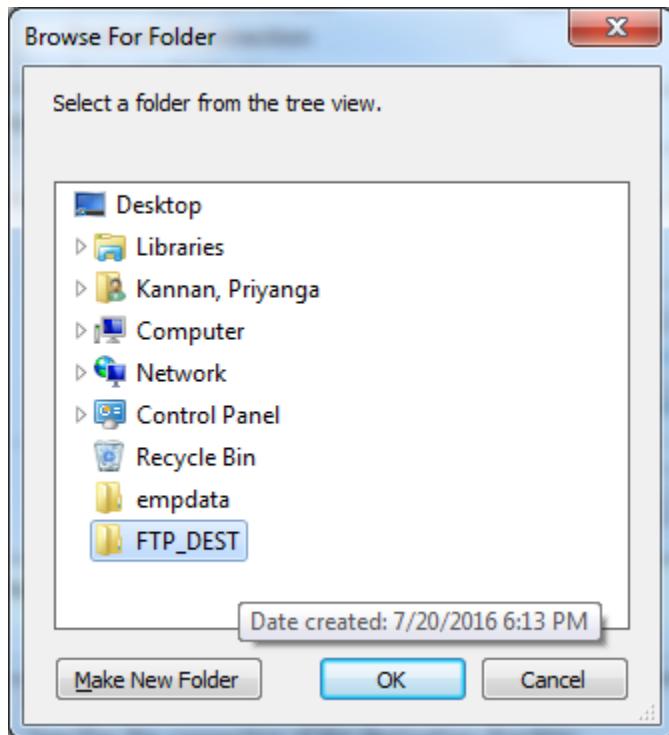


10)Now we have to configure the Destination Connection so, Please select the **DestinationConnection** property.

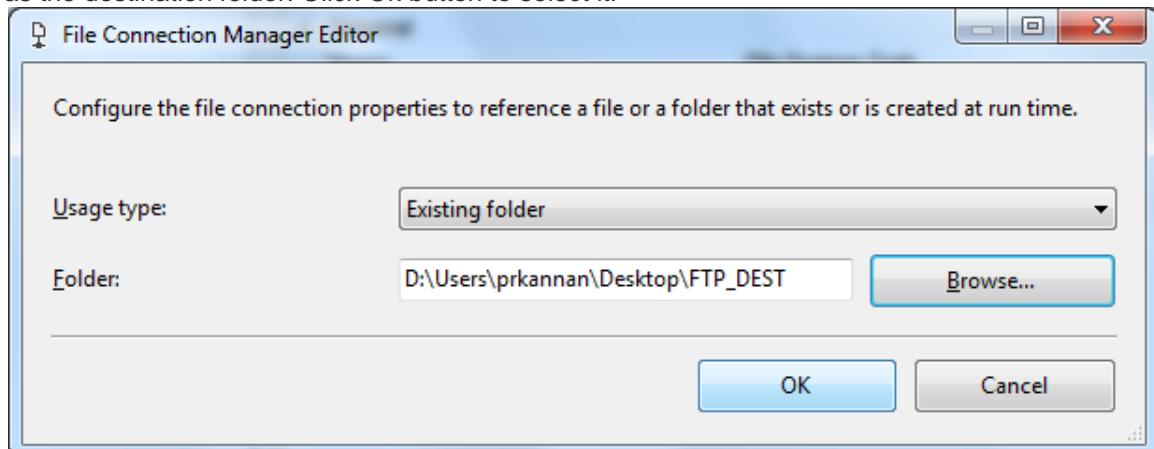
11)We haven't created any connection Manager before so, We are selecting <New Connection..>



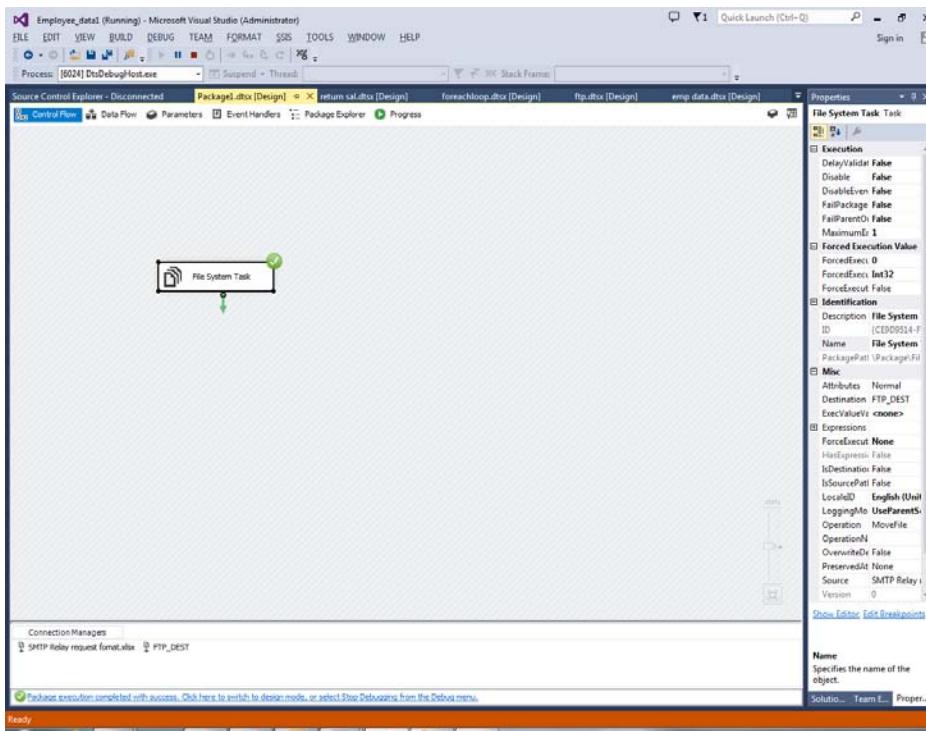
12)In this example we are Moving existing file to the already existing folder so, we are selecting Existing Folder option from the **Usage Type**.



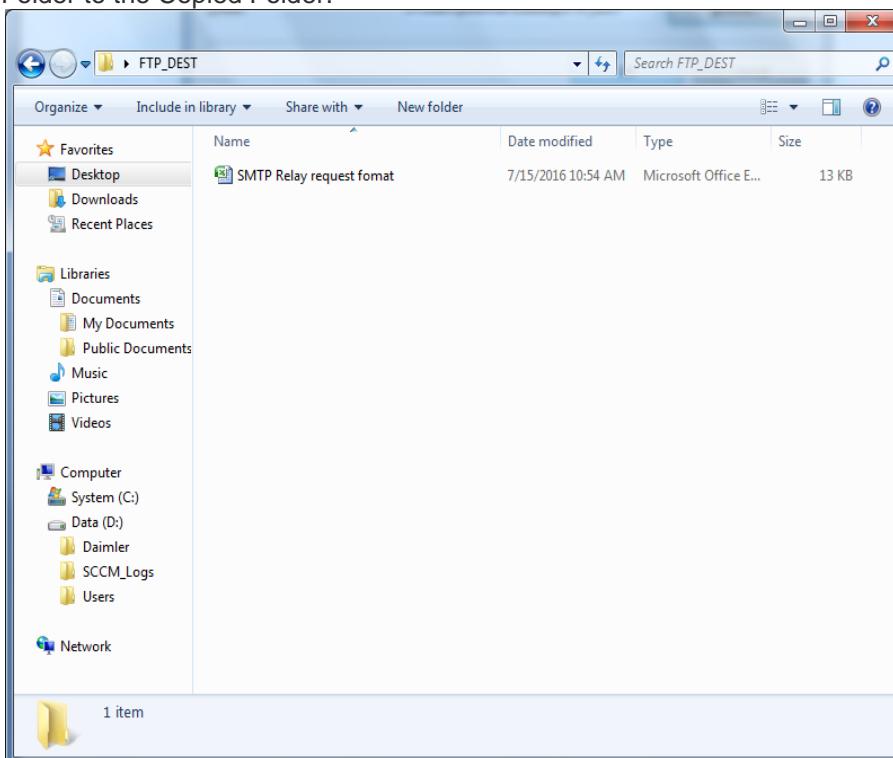
- 13) From the above the above screenshot you can observe that, We selected the Copied FTP_DEST as the destination folder. Click Ok button to select it.



- 14) Click Ok to finish configuring the File Connection Manager for the destination.
15) Click Ok to finish configuring Move File using File System Task in SSIS package.
16) Click Ok to finish configuring Move File using File System Task in SSIS package. Let's run and see whether we successfully Moved the Customers.txt file using the File System Task or Not.



17) We successfully Moved the SMTP Relay request format.xlsx file present in the File System Task Folder to the Copied Folder.



Lab 14-For each loop container with Execute SQL task and Script task

Objective	To learn <ol style="list-style-type: none"> 1. How to make use of control flow tasks 2. Use of For Each Loop container task as result set iterator 3. Calling Stored Proc 4. Using a Custom Script Task Component 5. Using DTS Variables
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Existing project and connection to database

- 1.Create a separate folder “Demo3”. Create a script CreateTab.sql and put the following T-SQL code in it:
2.Connect to SQL server using SSMS and create the following tables

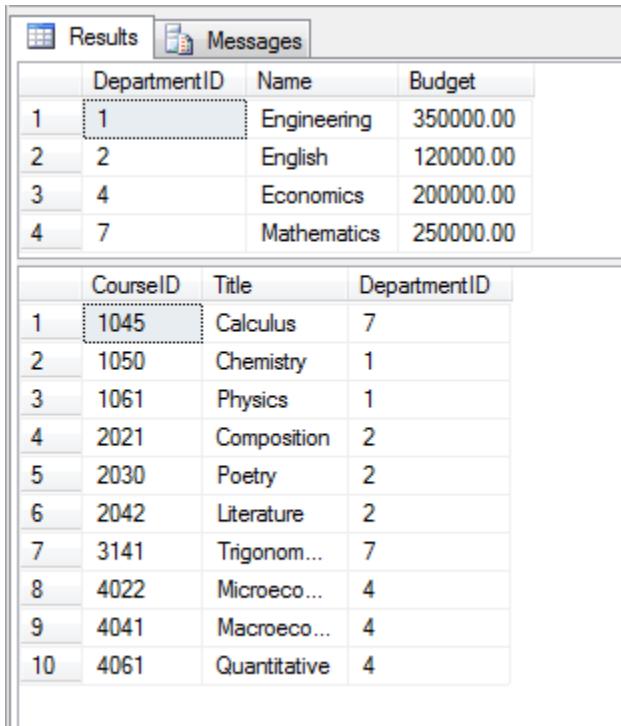
```

CREATE TABLE [dbo].[Department](
    [DepartmentID] [int] PRIMARY KEY,
    [Name] [nvarchar](50) NOT NULL,
    [Budget] [money] NOT NULL
)

CREATE TABLE [dbo].[Course](
    [CourseID] [int] PRIMARY KEY,
    [Title] [nvarchar](100) NOT NULL,
    [DepartmentID] [int] NOT NULL
)

```

- 3.Add the following records into the above tables:



	DepartmentID	Name	Budget
1	1	Engineering	350000.00
2	2	English	120000.00
3	4	Economics	200000.00
4	7	Mathematics	250000.00
	CourseID	Title	DepartmentID
1	1045	Calculus	7
2	1050	Chemistry	1
3	1061	Physics	1
4	2021	Composition	2
5	2030	Poetry	2
6	2042	Literature	2
7	3141	Trigonom...	7
8	4022	Microeco...	4
9	4041	Macroeco...	4
10	4061	Quantitative	4

4.Create a folder “Demo3”, and create a script “createproc.sql” in this folder. Put the following code in the script

```

ifOBJECT_ID('getcourses')isnotnull
dropprocedure GetCourses
go
createprocedure GetCourses
(
    @dno int,
    @coursecnt intoutput
)
as
    select @coursecnt =count(*)
        from course where departmentid = @dno
go

```

5.Add a new ssis package, Project => New SSIS package, name it “ExecuteSQLParametersResultSets.dtsx”.

6.In the control flow tab designer, drag a “Execute SQL Task”. Rename it as “Create Stored Proc”

a. double click task, set the following properties:

- i. connection type: oledb
- ii. sql source type: file connection.
- iii. connection: create new connection and connect to your sql server.
- iv. file connection: create a new connection which points to the “createproc.sql” script file created in step (4).

General	
Name	Create Stored Proc
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	pkplaptop_2285.ssisdemos
SQLSourceType	File connection
FileConnection	createproc.sql
IsQueryStoredProcedure	False
BypassPrepare	True

7. Select the package, by single clicking the package designer surface in free area, Open Variable Window, View => Other Windows => Variables

- b. Create a variable "DeptRS" clicking top left small button in the variables window, set datatype to "Object"
- c. Create 3 more variables Deptno Datatype Int32, DeptName datatype: string, CourseCnt datatype int32

Variables			
	Scope	Data Type	Value
X	DeptName	ExecuteSQLParametersResultSets	String
X	Deptno	ExecuteSQLParametersResultSets	Int32
X	DeptRS	ExecuteSQLParametersResultSets	Object
X	CourseCnt	ExecuteSQLParametersResultSets	Int32

8. Drag another "Execute SQL Task". Rename it as "Execute Department Query"

- d. Double click task, set the following properties:
 - i. ResultSet: Full Result Set
 - ii. connection type: oledb
 - iii. sql source type: direct input
 - iv. SQL Statement: SELECT * FROM department

General	
Name	Execute Department Query
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
Result Set	
ResultSet	Full result set
SQL Statement	
ConnectionType	OLE DB
Connection	pkplaptop_2285.ssisdemos
SQLSourceType	Direct input
SQLStatement	SELECT * FROM department
IsQueryStoredProcedure	False
BypassPrepare	True

9.Goto ResultSet TAB, click “Add” button and configure as below and click ok

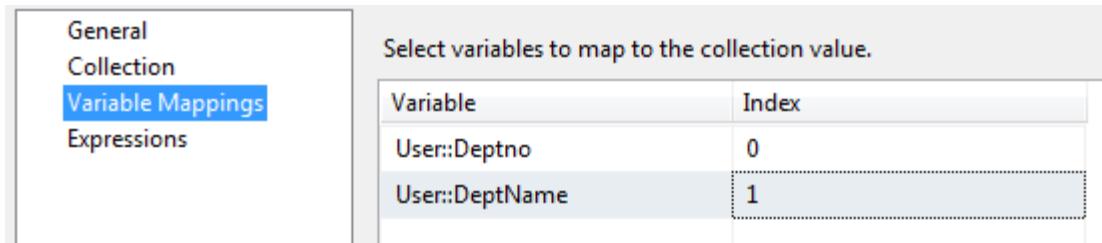
General	Result Name	Variable Name
Result Set	0	User::DeptRS

10.Drag a For Each Loop Container Task. Rename it “Iterate the Department Rows”

- e. Double click the task, goto collection tab, and set enumerator property to “ForEach Ado Enumerator”, set ado object source variable to User::DeptRS

General	Foreach Loop Editor
Collection	Enumerator
Variable Mappings	Foreach ADO Enumerator
Expressions	Expressions
Enumerator	
Specifies the enumerator type.	
Enumerator configuration	
ADO object source variable:	
User::DeptRS	

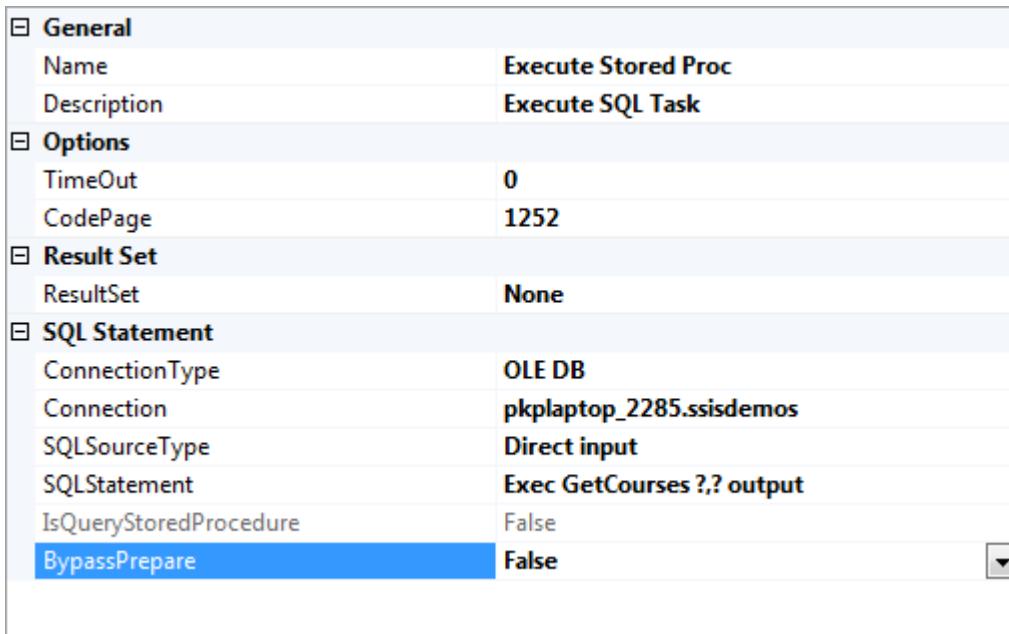
11.Goto Variable Mappings tab and configure as below and then click Ok



Variable	Index
User::Deptno	0
User::DeptName	1

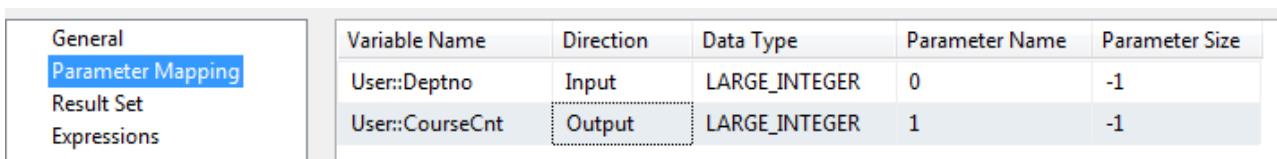
12. Drag a Execute SQL Task inside the For Each Loop Container, Rename the task as "Execute Stored Proc"

- f. Double click task, set the following properties:
 - i. ResultSet: None
 - ii. connection type: oledb
 - iii. connection: specify a connection to your sql server
 - iv. sql source type: direct input
 - v. SQL Statement: Exec GetCourses ?,? output
 - vi. ByPass Prepare: false



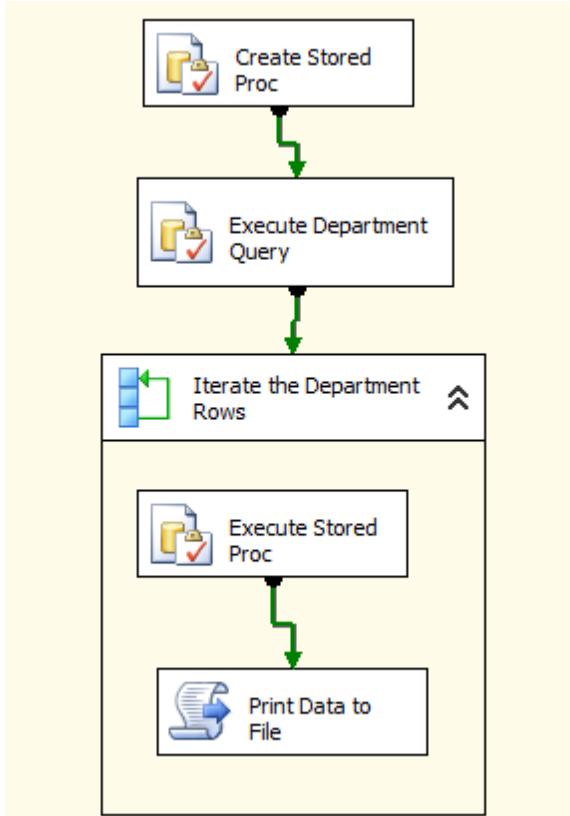
Name	Execute Stored Proc
Description	Execute SQL Task
TimeOut	0
CodePage	1252
ResultSet	None
ConnectionType	OLE DB
Connection	pkplaptop_2285.ssisdemos
SQLSourceType	Direct input
SQLStatement	Exec GetCourses ?,? output
IsQueryStoredProcedure	False
BypassPrepare	False

2. Goto Parameter Mapping Tab and configure it as below and click Ok
Note: Click Add button to add variable entries as parameters



Variable Name	Direction	Data Type	Parameter Name	Parameter Size
User::Deptno	Input	LARGE_INTEGER	0	-1
User::CourseCnt	Output	LARGE_INTEGER	1	-1

3. Drag a script task inside the For Each Loop container task and rename it as "Print Data to File"
4. Connect the various tasks that are dragged sequentially as shown below:



5. In the package designer right click the “Connection Managers” section at the bottom in free space and select “new file connection”
 - a. Set Usage Type: Create File and
 - b. Browse and specify filename as Results.txt in Demo3 folder created at the start of this lab. Click ok
6. Double the script task and in the ellipses for readonly variables property, select User::CourseCnt, User::DeptName, User::Deptno Variables.
7. Click Edit Script button
8. Define the following code in the main function

```

publicvoid Main()
{
    int? dno,coursecnt;
    string dname;

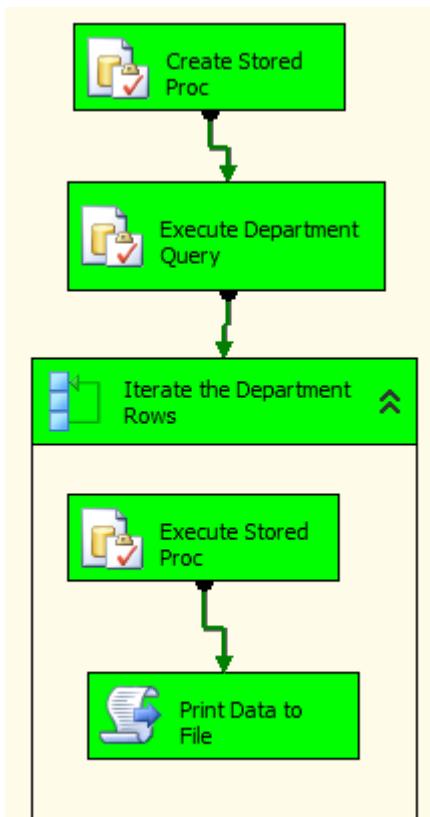
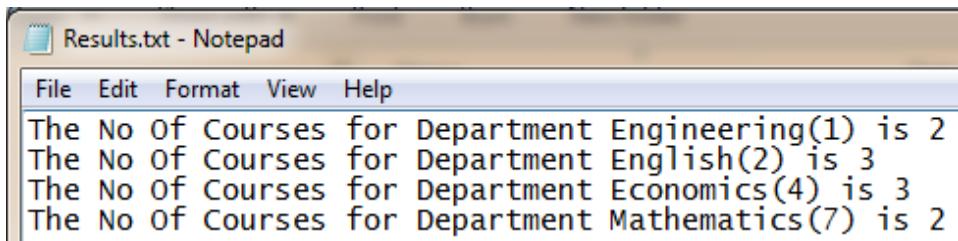
    string FileName = (string)Dts.Connections["Results.txt"].AcquireConnection(null);
    dno = (int)Dts.Variables["Deptno"].Value;
    if (Dts.Variables["CourseCnt"].Value == null)
        coursecnt = 0;
    else
        coursecnt = (int)Dts.Variables["CourseCnt"].Value;
    dname = (string)Dts.Variables["DeptName"].Value;
    using (StreamWriter sw = File.AppendText(FileName))
    {
  
```

```

        sw.WriteLine("The No Of Courses for Department " + dname + "(" +
            dno.ToString() +") is " + coursecnt.ToString());
        sw.Close();
    }
    Dts.TaskResult = (int)ScriptResults.Success;
}

```

9. Build the script Ctrl+Shift+B. Once succeeds exit the script editor. Click ok on the script task dialog box to go to the package designer.
10. Run the package and open the Results.txt file as shown below and then stop the package



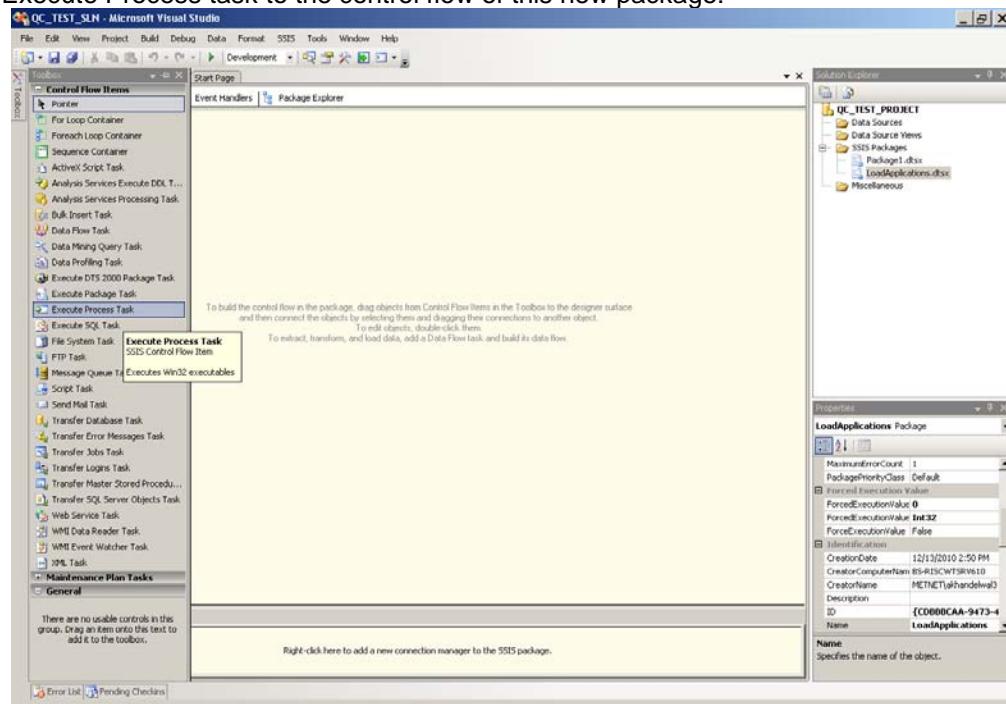
Open a different application based on day of week

Lab 15- Execute Process Task

Objective	The purpose of this Lab is to show an example of property expressions in a fringe operation, using the Execute Process task to open other applications.
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Create package.

1)Save and Close all other open packages and create a new package as shown below and then rename it:

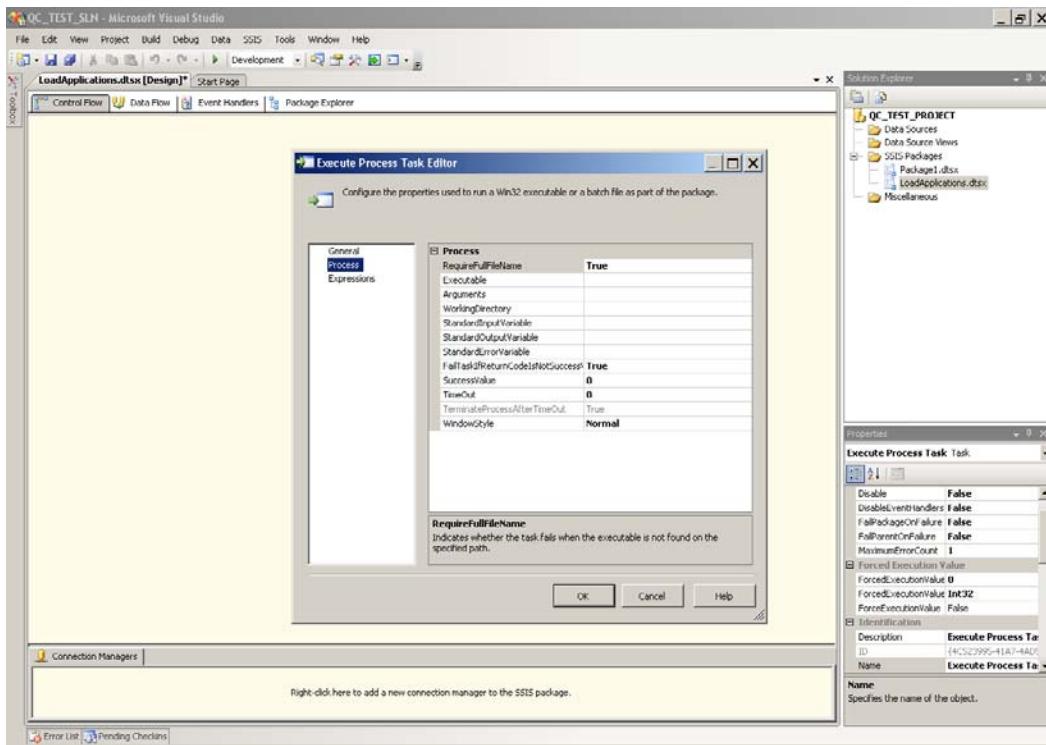
2)Add an Execute Process task to the control flow of this new package:



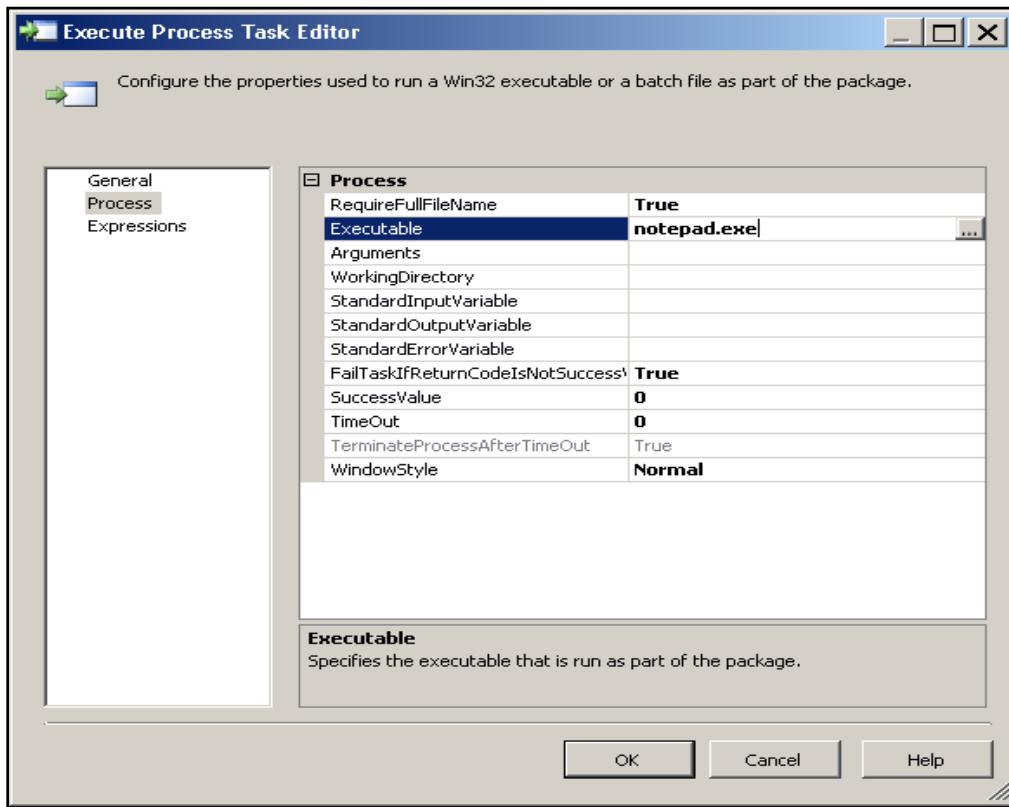
Note:

[Using property expressions we can configure the single task to open a different application based on the day of the week. Open either notepad.exe or mspaint.exe depending on day of week.]

3)Now double click on the Execute Process task to go to editor and click on the Process Page:

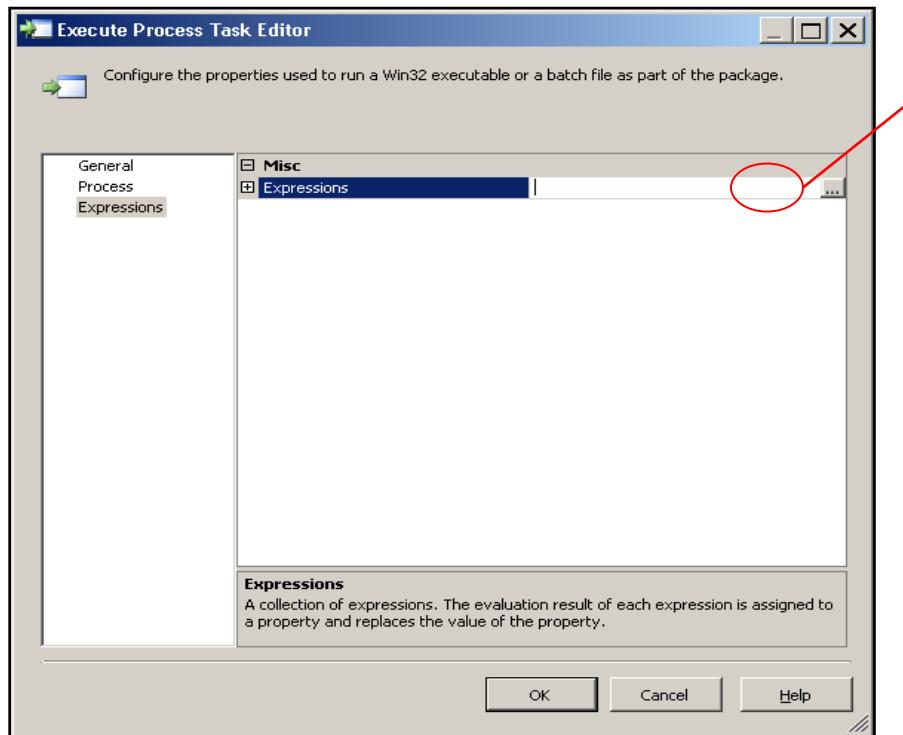


4)Now enter “notepad.exe” for the ‘Executable’ property.



5)Now click the “Expressions” page. In the right pane click in the empty row for “Expressions” and then press the ellipse button which will take you to the Property Expressions Editor.

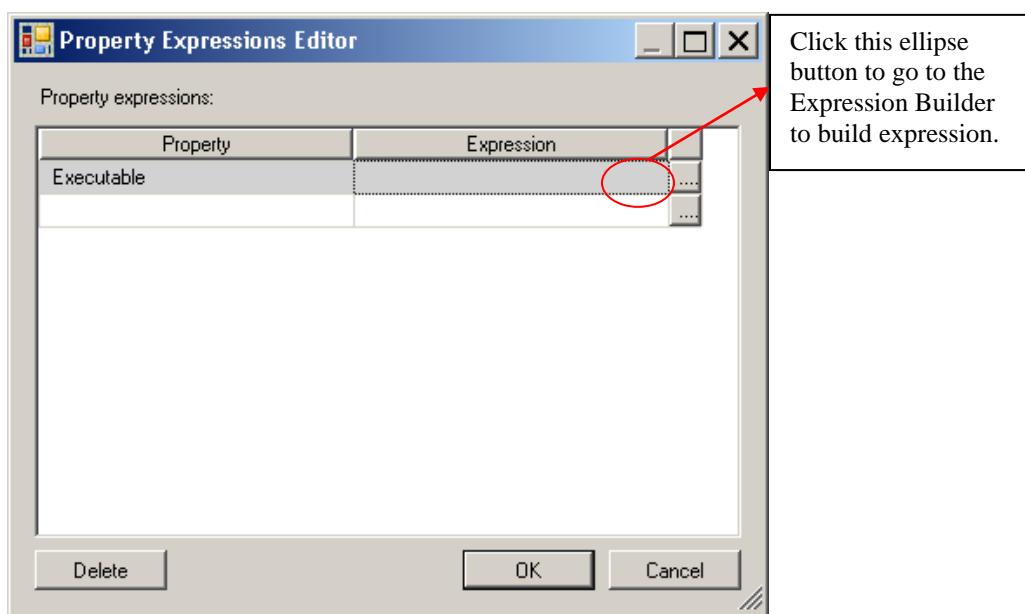
Press this
ellipse
button



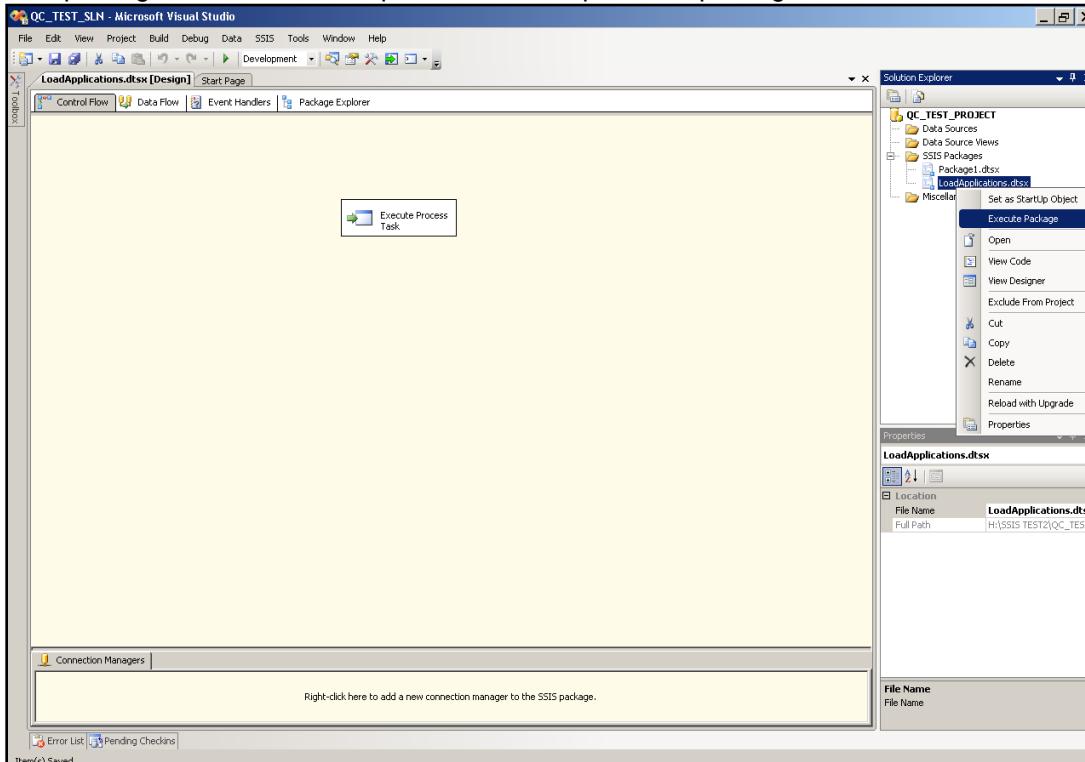
6)Now choose the “Executable” property and either copy/paste the following expression or press the other ellipse button to go into the expression builder and build this yourself and then evaluate the expression:

Expression:

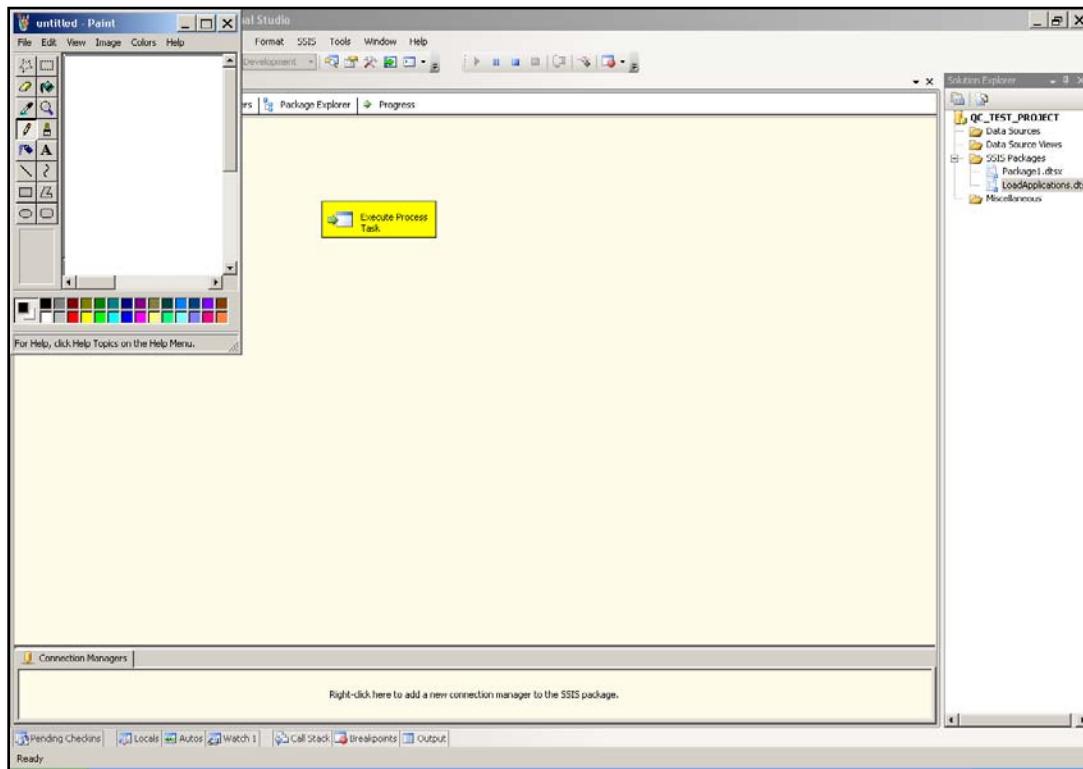
DATEPART("weekday", GETDATE()) ==5?"notepad.exe":"mspaint.exe"



7)Save the package. In the solution explorer select this particular package and select “Execute Package”.

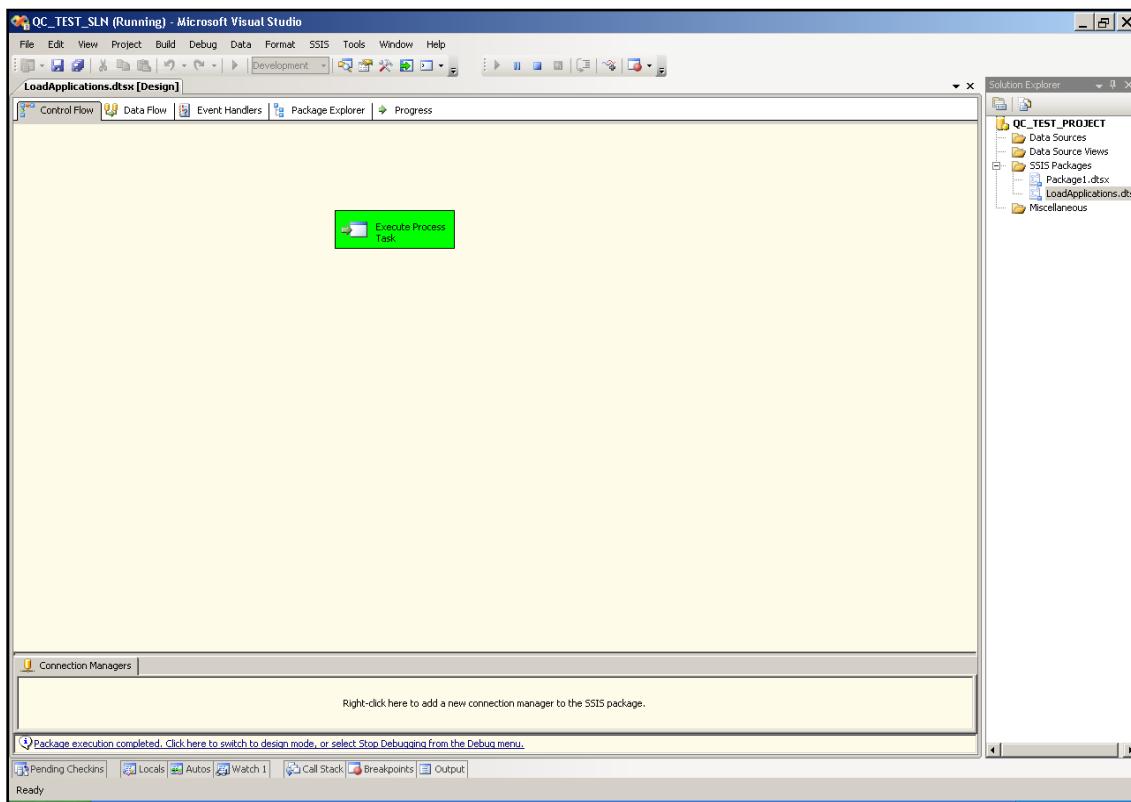


8)IF the current day is Thursday (5) then Windows Notepad will open, otherwise Paint will open as shown below:



Note: The Execute Process task is still yellow and the package is still considered 'running' until you close the application that opened, then the task will turn green.

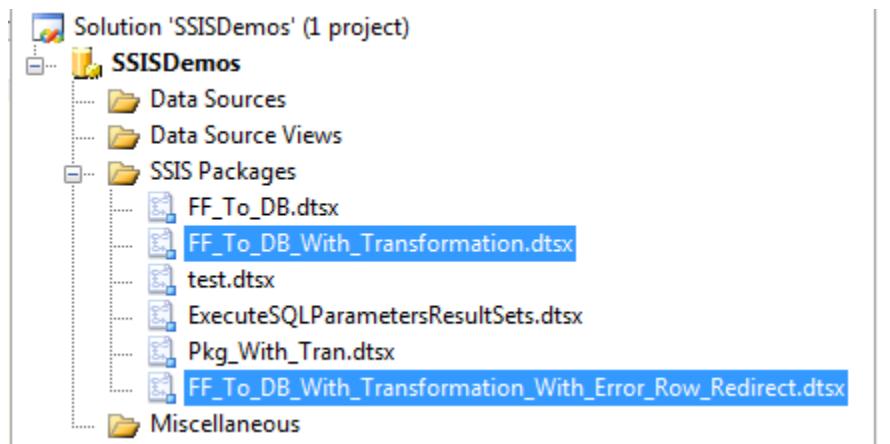
9) After closing the Paint application this task turned green.



Lab 16- Using Error Redirect Row

Objective	To learn 1. How to use error redirect output to identify the rows having processing errors and prevent the entire ELT process from halting
Lab Setup	<ul style="list-style-type: none"> SSDT tool Login details for connecting to the database and Create package.

1)In Solution Explorer copy and paste the “FF_To_DB_With_Transformation.dtsx” and rename as “FF_To_DB_With_Transformation_With_Error_Row_Redirect.dtsx”. Click “Yes” to the dialog box that will come up to complete renaming.



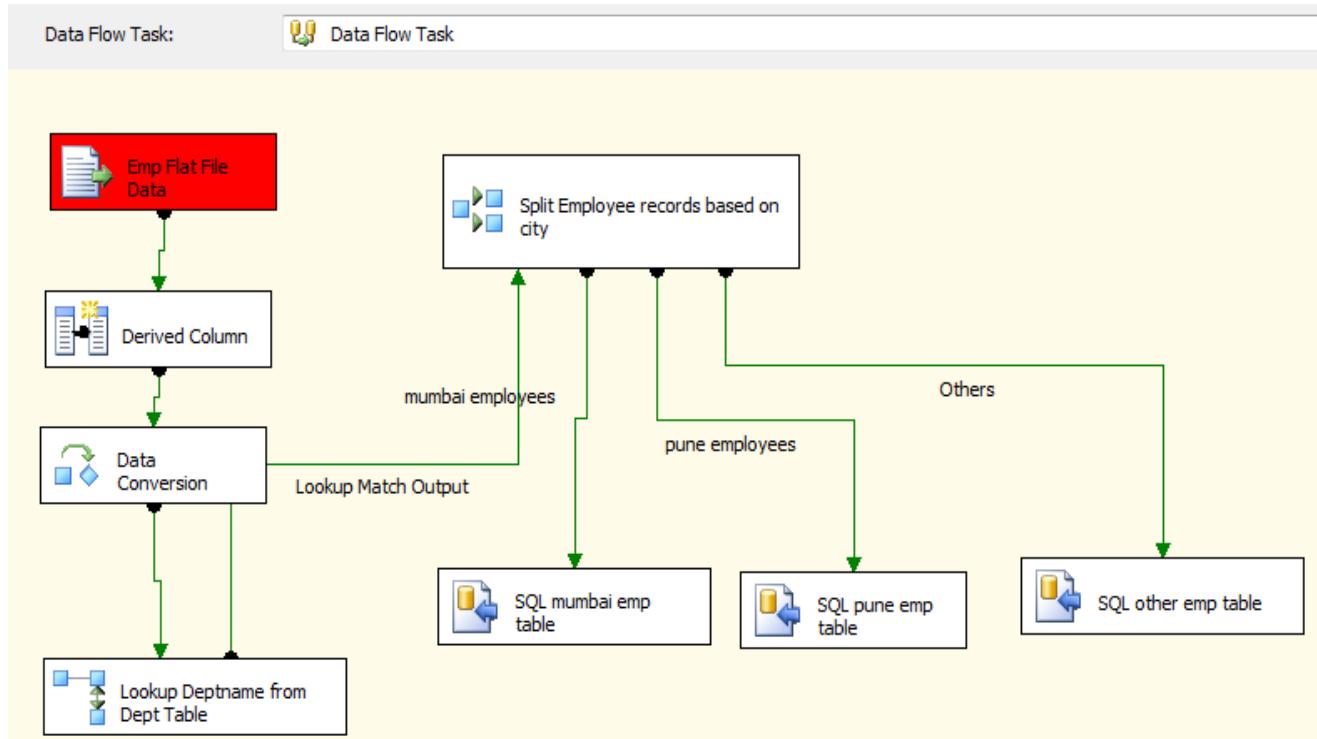
2)Goto Demo2 folder created for Lab 2 open “emp1.txt” file and change the salary value of one the employee to character as shown and save changes

```
e101,bill gates,mumbai,70000,10
e102,ajit jog,mumbai,50000,10
e103,sachin tendulkar,pune, 40000,20
e104,sunil gavaskar,, 60000,20
e105,zakir hussain,mumbai, abcd,10
e106,akshay,delhi, 110000,10
```

3) Prior delete all the records from the tables

```
deletefrom mumbaiemp
deletefrom puneemp
deletefrom otheremp
```

4) Open “FF_To_DB_With_Transformation.dtsx” package and execute the package. There will be error in the package execution as shown:



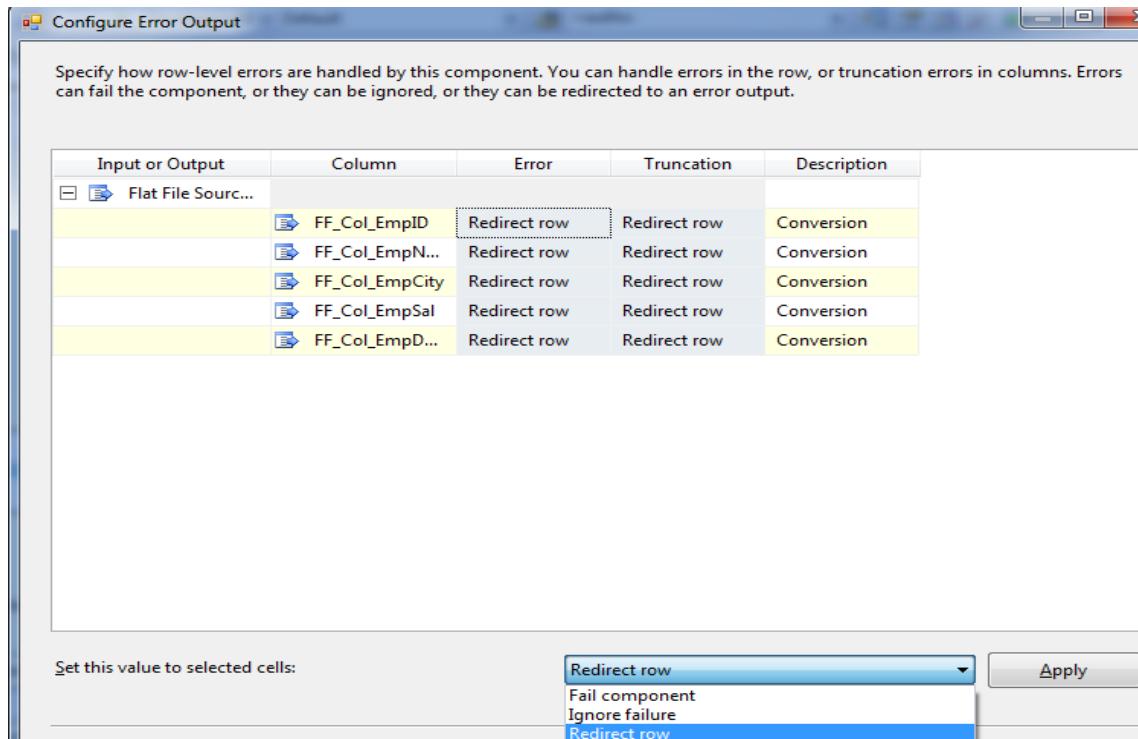
5) Stop the Package execution from VS Studio.

6) Check the SQL tables, no data would be transferred.

7) Open “FF_To_DB_With_Transformation_With_Error_Row_Redirect.dtsx” goto data flow tab, from toolbox drag “Flat File Destination” task.

8) Join the Red Error Output Connector of the “Emp Flat File Data” Flat File Source Task to it.

9) In the dialog box select all the error and truncation column entries. (Use: “shift” + click) select “redirect row” from dropdown below and click appl. Click ok

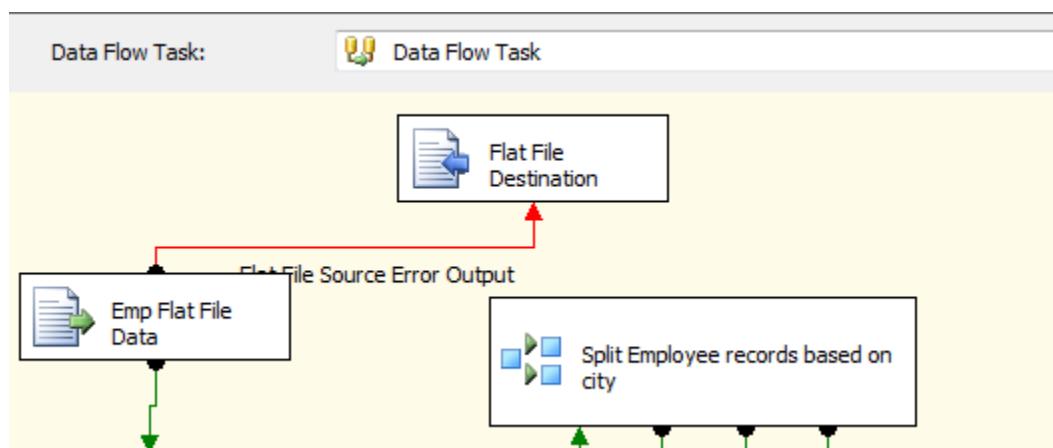


10) Double Click the “**Flat File Destination**” task click new “New” button to create a new Connection Manager. Select “Delimited” in the dialog box that pops. Name the connection manager as “FFCM_Error_Output_File”.

11) In filename specify the file as “emperr.txt” in the same directory Demo2

12) For eg: C:\Users\jogajitm\Desktop\Project 1\Folders\Demo2\emperr.txt

13) Click Ok.



14) Save Changes to package. Run it.

15) The package will succeed. Stop it. The error row will come into the emperr.txt file rest of the rows will be successfully transferred to the SQL tables.

Lab 17- Transaction support and Checkpoint

Objective	To learn <ul style="list-style-type: none"> 1. How to use Transaction support available for atomicity 2. How to use Checkpoint feature to resume the package from the point where it was abandoned
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Login details for connecting to the database and Create package.

1)Create Demo5 folder for this lab. Create “In”, “Out” and “Checkpoint” 3 folders under it.

2)Connect to sql server and create 2 tables as below:

```

CREATE TABLE staginginvoices
(
    invno int PRIMARY KEY,
    invamt numeric(18, 0) NULL,
    invdate smalldatetime NULL,
)

CREATE TABLE invoices
(
    invno int PRIMARY KEY,
    invamt numeric(18, 0) NULL,
    invdate smalldatetime NULL,
)

```

Add few records into staging invoices table for eg as below:

Results			Messages	
	invno	invamt	invdate	
1	1	10000	2012-01-01 00:00:00	
2	2	25000	2012-01-03 00:00:00	
3	3	40000	2012-01-03 00:00:00	

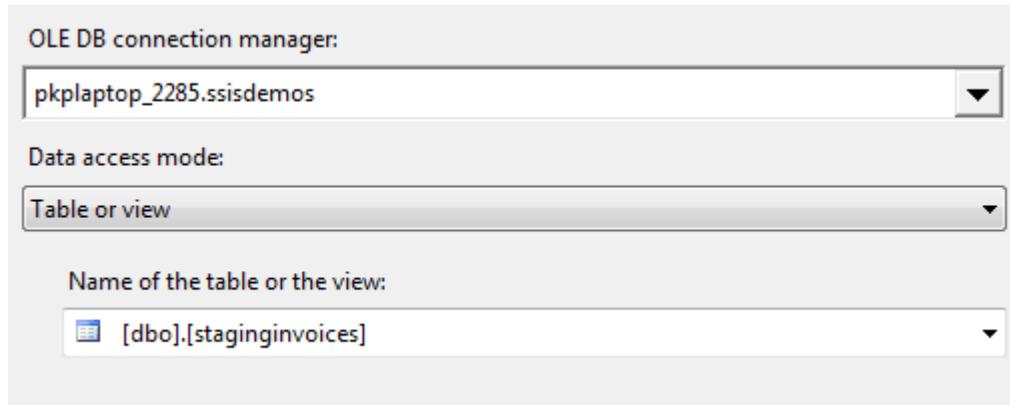
3)Add a new ssis package and name it as “**Pkg_with_Tran_and_CheckPoint_Demo.dtsx**”.

4)In the control flow, drag a sequence container task. Rename it “Transfer Invoices to Table and File”.

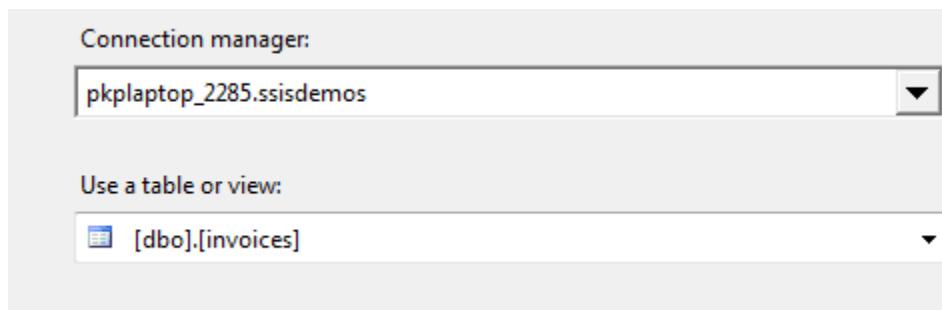
5)Drag a data flow task inside this sequence container. Rename data flow task to “Transfer Invoices to SQL Table”.

6)double click the above data flow task.

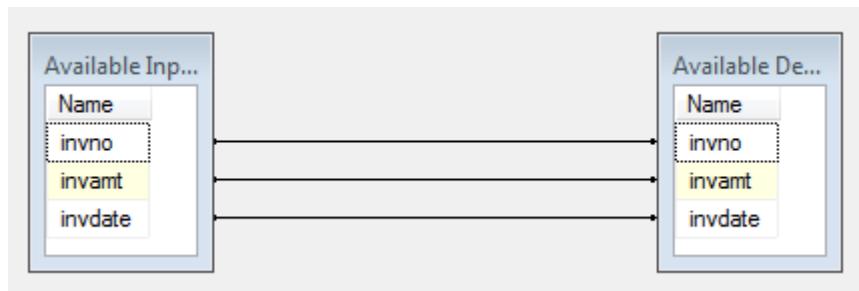
7)Drag a oleodb data source task from data flow sources section. Rename as “Staging Table” Double click and point it to the sql server database where staginginvoices table is created as shown below. Click ok



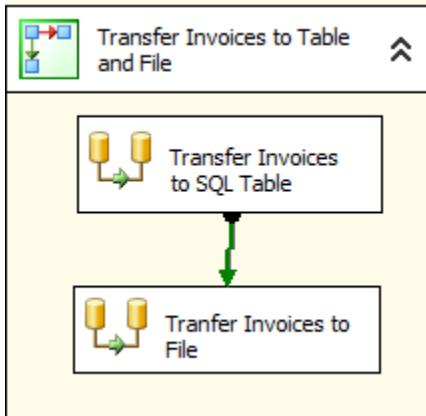
8) Drag a Sql server destination task. Rename it “Invoices Table”. Connect the oledb datasource task to the destination task. Double click destination task and point it to invoices table created above.



9) Go to mappings tab and map the corresponding columns. Click ok



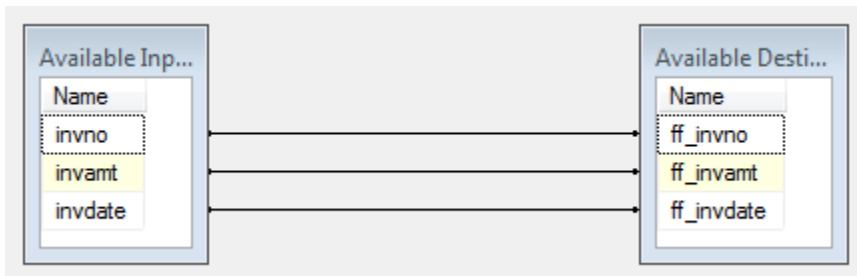
10) In control flow, drag another data flow task in the sequence container, below the earlier data flow task and connect them. Rename this task as “Transfer Invoices to File”.



- 11) Double click above data flow task to invoke data flow designer,
- 12) Drag a oledb data source and point it to staginginvoices.
- 13) Drag a flat file destination, rename it "Invoices Flat File".
- 14) Connect the oledb source of step 12 to it. double click flat file destination and through a new flat file connection point it to a file in "In" folder of step 1.

For eg: C:\Users\jogajitm\Desktop\Project 1\Folders\Demo5\In\invoices.txt

- 15) Set the mappings as below:



- 16) Back to control flow, drag a Execute SQL Task outside sequence container, rename as "Empty Staging table". Double click it and configure it as below:

General	
Name	Empty Staging table
Description	Execute SQL Task
Options	
TimeOut	0
CodePage	1252
Result Set	
ResultSet	None
SQL Statement	
ConnectionType	OLE DB
Connection	pkplaptop_2285.ssisdemos
SQLSourceType	Direct input
SQLStatement	delete from staginginvoices
IsQueryStoredProcedure	False
BypassPrepare	True

Note: Connection: is the connection manager which points to staginginvoices sql server database.

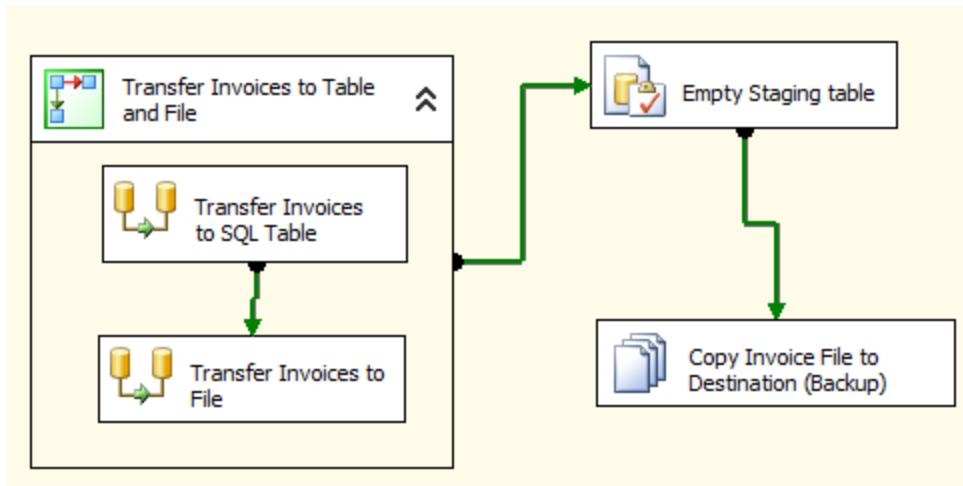
- 17) Drag a File System task on to control flow and rename as "Copy Invoice File to Destination (Backup)".
18) Double click the above task and configure as below:

Destination Connection	
IsDestinationPathVariable	False
DestinationConnection	invoices.txt
OverwriteDestination	False
General	
Name	Copy Invoice File to Destination (Backup)
Description	File System Task
Operation	
Operation	Copy file
Source Connection	
IsSourcePathVariable	False
SourceConnection	FF_Invoices_CM

Note:

- **Destination connection** is a flat file connection manager which points to some file in "Out" folder created in step 1.
For eg: C:\Users\jogajitm\Desktop\Project 1\Folders\Demo5\Out\invoices.txt
- **Source Connection** is the flat file manager created above in step 14 which points to the file in "In" folder
- The **operation property** is set to Copy File.

19) Connect the control flow tasks as shown below



Configure Transaction Support:

- 20) Select the sequence container task, and set the transaction option of the task to “required”.
21) Set the transaction option of both the data flow tasks within sequence container to “supported”

Configure Checkpoint Support:

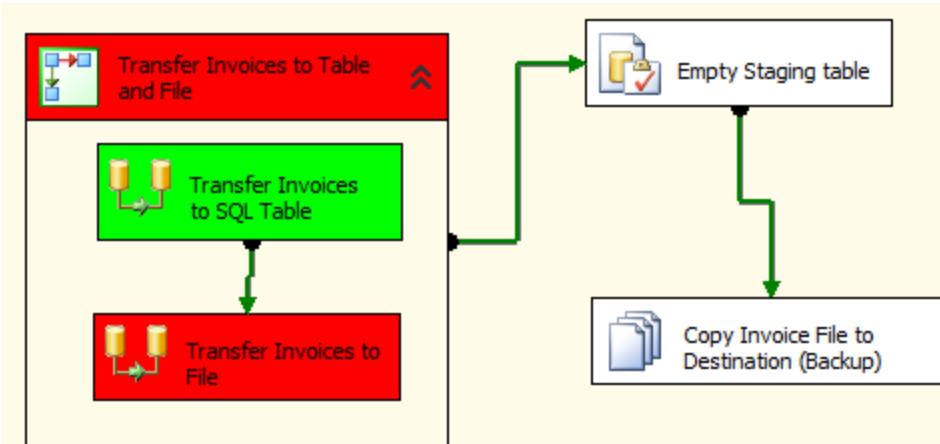
- 22) Select the control flow designer , by clicking it on free surface.
23) Set the following properties

Property Name	Value
Checkpoint File Name	To a file name in checkpoint folder created in step 1 For example something like: C:\Users\jogajitm\Desktop\Project 1\Folders\Demo5\CheckPoint\ChkPnt.txt
CheckPoint Usage	If Exists
Save Checkpoints	True

- 24) Make use that for all tasks in the control flow viz: sequence container, sub tasks of sequence container, execute sql task, file system task, the properties “FailPackageOnFailure” and “FailParentOnFailure” properties are true.

→ Now we ready to run the package and test the features

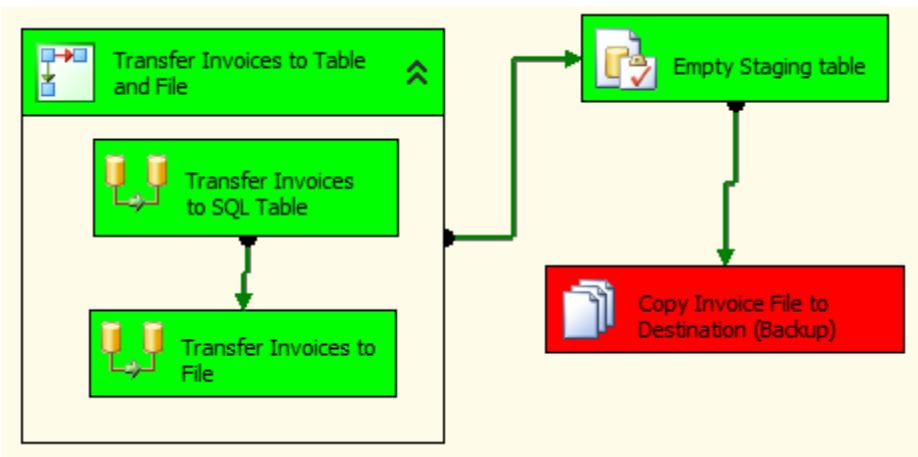
- 25) Before we run delete the “In” folder for time being or rename it to something else.
26) Run the Package, the package will fail in the sequence container because “In” folder is not found while transferring invoices details to flat file as below:



27) Go to SSMS and check that the work of first data flow task is rolled back. The data is not successfully transferred to Invoices online table from staging table. Stop the package

28) Now recreate the “In” folder, but now delete the “Out” folder from Demo5 folder for the time being.

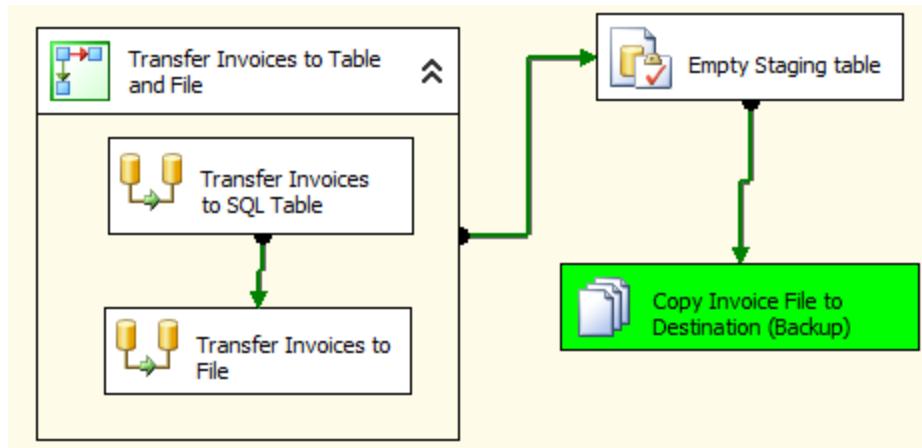
29) Now run the package, the package fails at the File System Task “Copy Invoice File to Destination (Backup)” but the earlier tasks are completed as below:



30) Stop the package, and check that:

- Rows have been copied to invoices table from staging table.
- The flat file is created in “In” folder
- A checkpoint has been created in checkpoint folder

31) Stop the package and restore or recreate the “Out” folder and rerun the package.



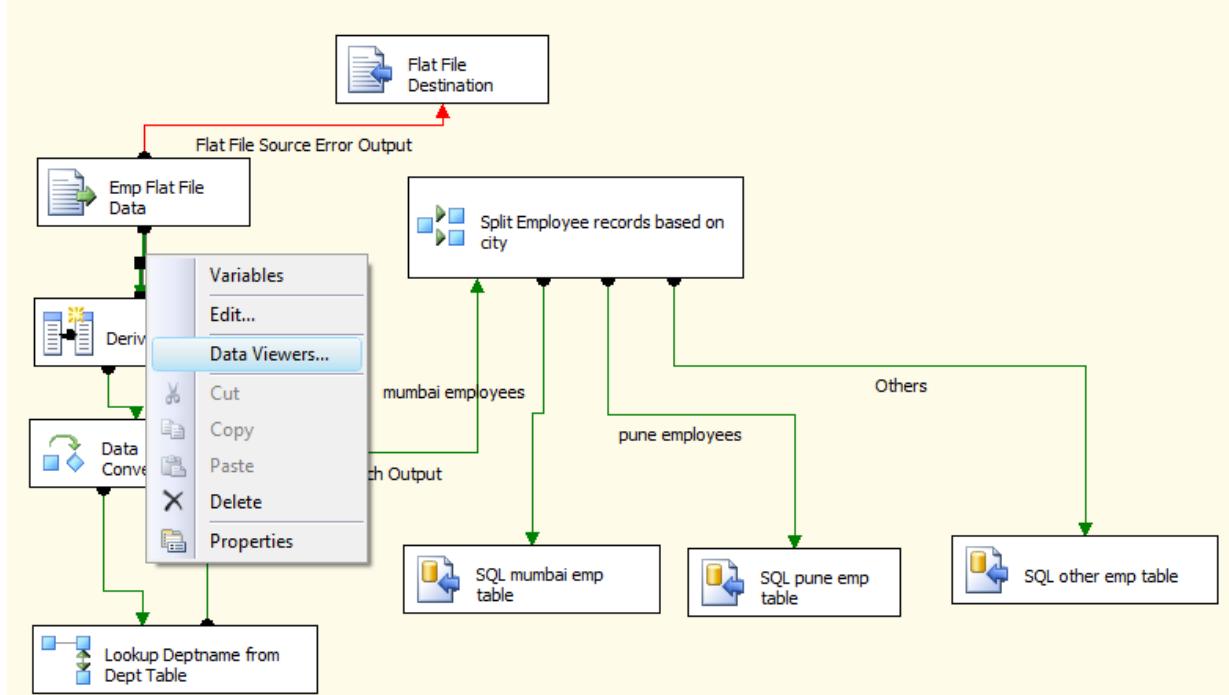
- 32)The package will run directly run the last file system task and the checkpoint file will be automatically removed from its location and the file will be copied to the “Out” folder.
33)Stop the package.

Lab 18- Using Data Viewer

Objective	To learn 1. How to use data viewer
Lab Setup	<ul style="list-style-type: none"> SSDT tool Create package.

1)Open the Package "FF_To_DB_With_Transformation_With_Error_Row_Redirect.dtsx" created in Lab 4

2)Right Click the data flow path between the "Emp Flat File Data" and "Derived Column" data flow tasks.



3)Click Data Viewers

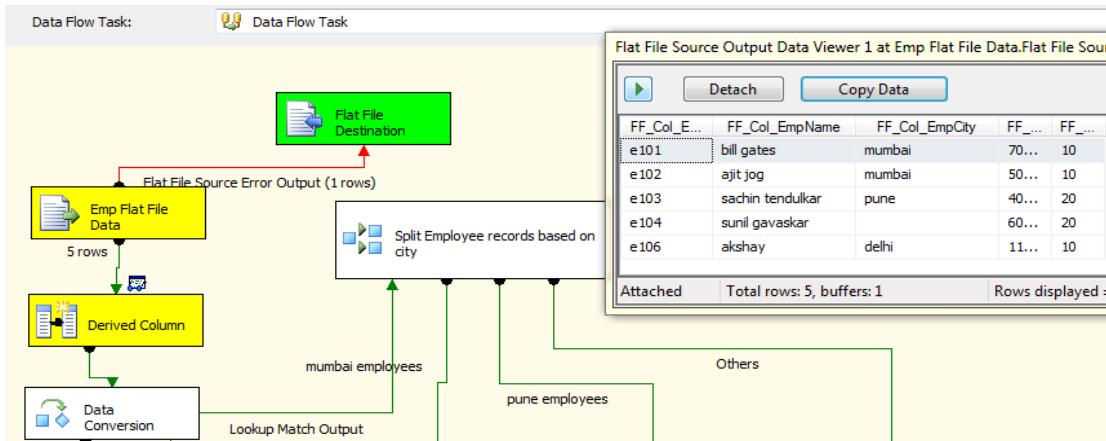
4)In the Data Viewer dialog box click "Add" button.

5)Select Grid from options. Click ok. Again click Ok.

6)delete the data from mumbaiemp, puneemp and otheremp tables.

7)Execute the package.

8)A data viewer window will automatically popup and will show the records retrieved from the flat data file. The package execution will also pause for a moment.



9) Resume the package execution click  button on data viewer window and complete the execution.

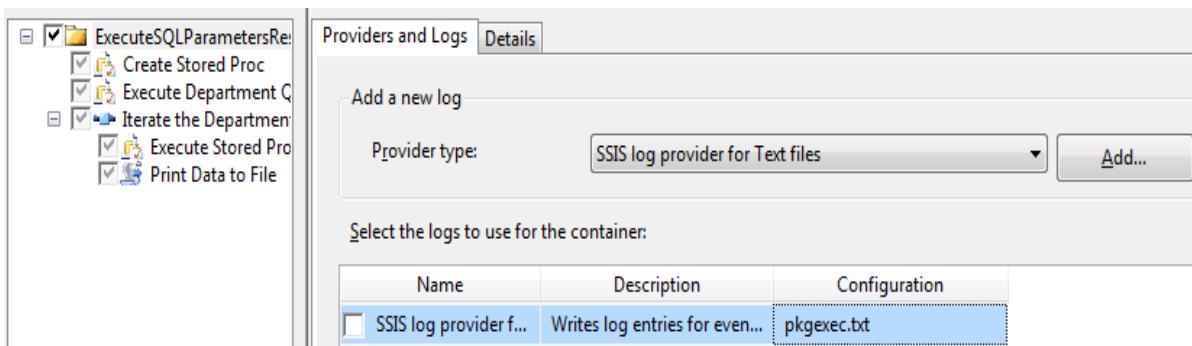
Lab 19- Using Package Logging

Objective	To learn <ol style="list-style-type: none"> 1. How to enable package logging
Lab Setup	<ul style="list-style-type: none"> • SSDT tool • Create project.

1)Open the Package “ExecuteSQLParametersResultSets.dtsx” of Lab 3.

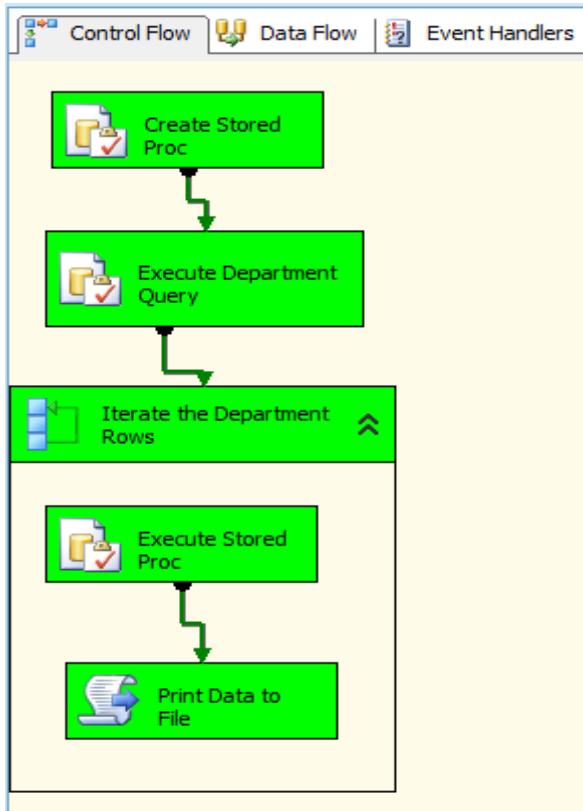
2)To enable logging for this package go to SSIS => Logging Menu option.

- In the logging dialog box, select the root tree node of the package in left pane.
- In right pane, select “SSIS log provider for text file” and click add. A row will be added.
- Go to configuration column of the row, Click new connection.
- Select Create File, and give a file path, where you want your package execution log file to be created.
- Click ok



- a. Now go to details tab, select “onpreexecute”, “onpostexecute” and “ontaskfailed ”events.
- b. Click ok.

3)Execute the package.



4) Stop the execution on the package.

5) Open the package log file. The package file contents will look as below:

```

pkexec.txt - Notepad
File Edit Format View Help
#Fields: event,computer,operator,source,sourceid,executionid,starttime,endtime,datacode,databytes,message
PackageStart,PKPLAPTOP_2285,IGATECORP\jogajitm,ExecuteSQLParametersResultSets,{941EED59-313A-46D4-9982-5D48A71AF57F},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,0x,Beginning of package execution.

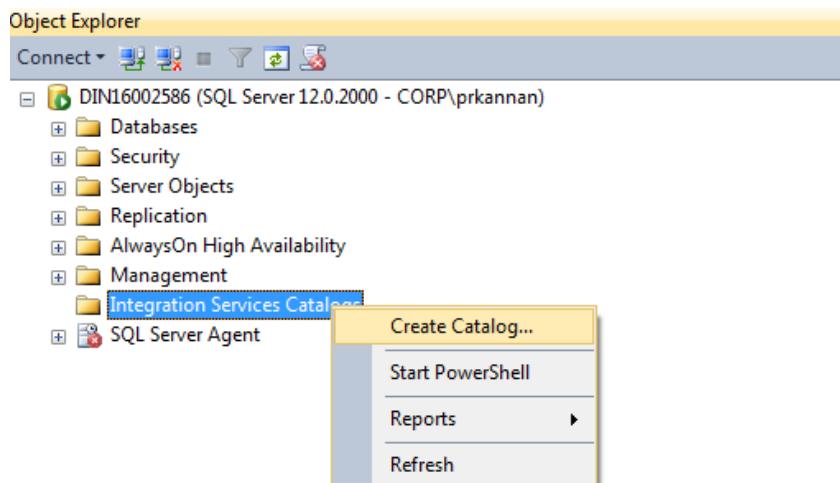
OnPreExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,ExecuteSQLParametersResultSets,{941EED59-313A-46D4-9982-5D48A71AF57F},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPreExecute,PKPLAPTOP_2285,IGATECORP\jogajitm>Create Stored Proc,{25BC5D01-0292-49C3-B010-B397BE6553D9},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPostExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Create Stored Proc,{25BC5D01-0292-49C3-B010-B397BE6553D9},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPreExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Execute Department Query,{6F7E51B9-9861-439D-A6EE-0154983B8215},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPostExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Execute Department Query,{6F7E51B9-9861-439D-A6EE-0154983B8215},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPreExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Iterate the Department Rows,{7857942A-D374-4E89-B8D8-57E83FA3715F},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPostExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Execute Stored Proc,{38EB36EA-461F-433E-83D5-BAB39F28FECC},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
OnPostExecute,PKPLAPTOP_2285,IGATECORP\jogajitm,Execute Stored Proc,{38EB36EA-461F-433E-83D5-BAB39F28FECC},{D3FEFF66-6C59-4520-8E4F-5AD85A5C1643},27-01-2012 11:11:43,27-01-2012 11:11:43,0,(null)
  
```

Lab20-Project deployment

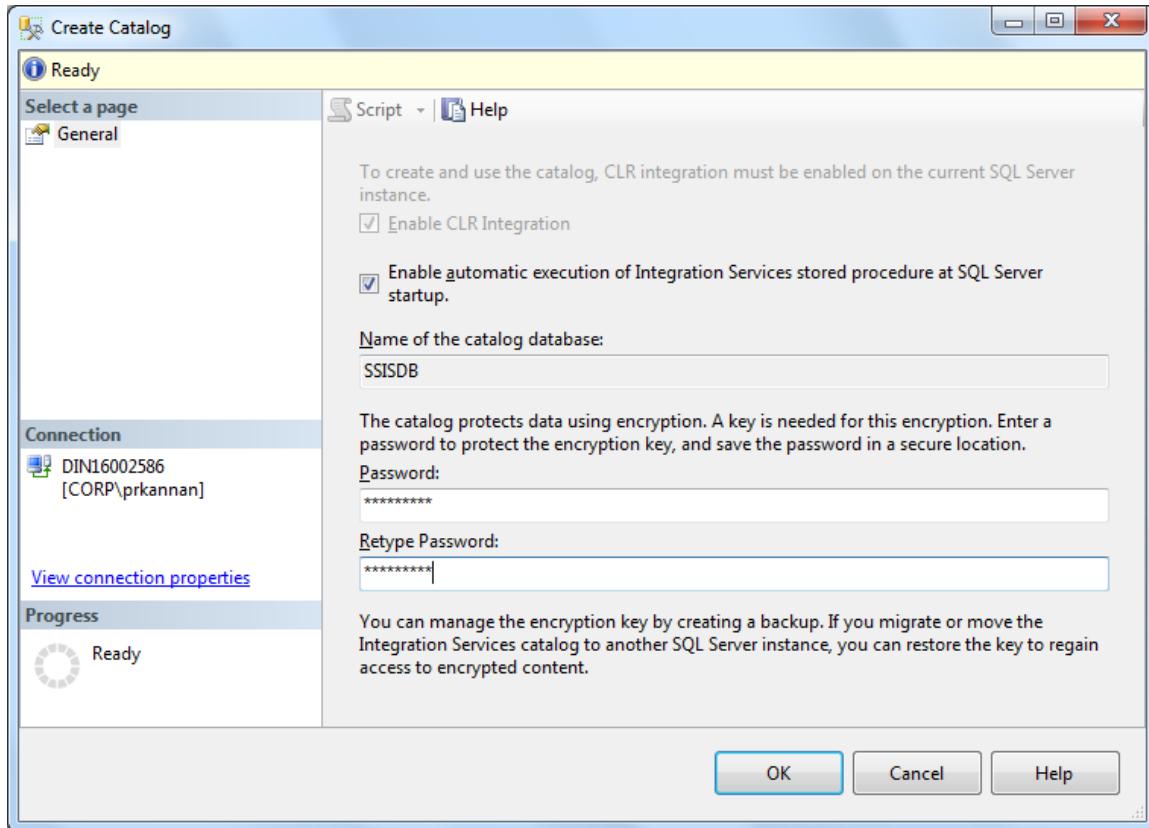
Objective	To learn 1.How to deploy a project.
Lab Setup	<ul style="list-style-type: none"> SSDT tool Login details for connecting to the database

IN SSMS:

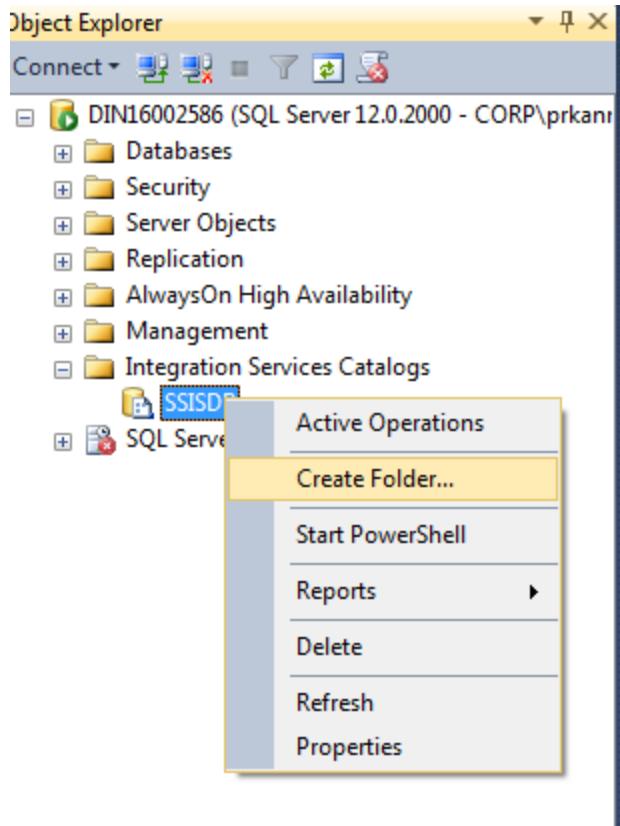
- 1) Before we can deploy any projects to a SQL Server instance, we first need to create the Integration Services Catalog. Think of this catalog as the container for all deployed projects, their settings and historical versions.
- 2) There is only one Catalog per SQL Server instance; it is represented by a separate SQL Server database called SSISDB, which contains the deployment's versioning, settings, and even statistical performance data.
- 3) In order to create the catalog, we need to connect to the SQL Server instance and right-click on the 'Integration Services Catalogs' and click 'Create Catalog'.



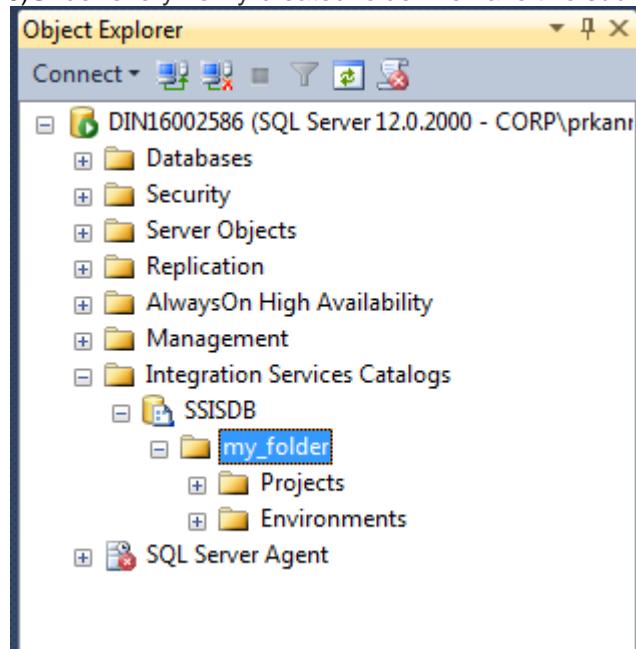
- 4) In the next screen we are asked for the encryption password and whether we would like to run the **Catalog.Startup** procedure every time the SQL Server instance starts (the **Catalog.Startup** stored procedure fixes the status of the packages in the catalog if there were packages running when the SQL Server instance went down):



5) After we have created the Catalog, we can create a new folder under it, which will contain our first project. To create the folder, simply right-click on the SSISDB catalog and click 'Create Folder':

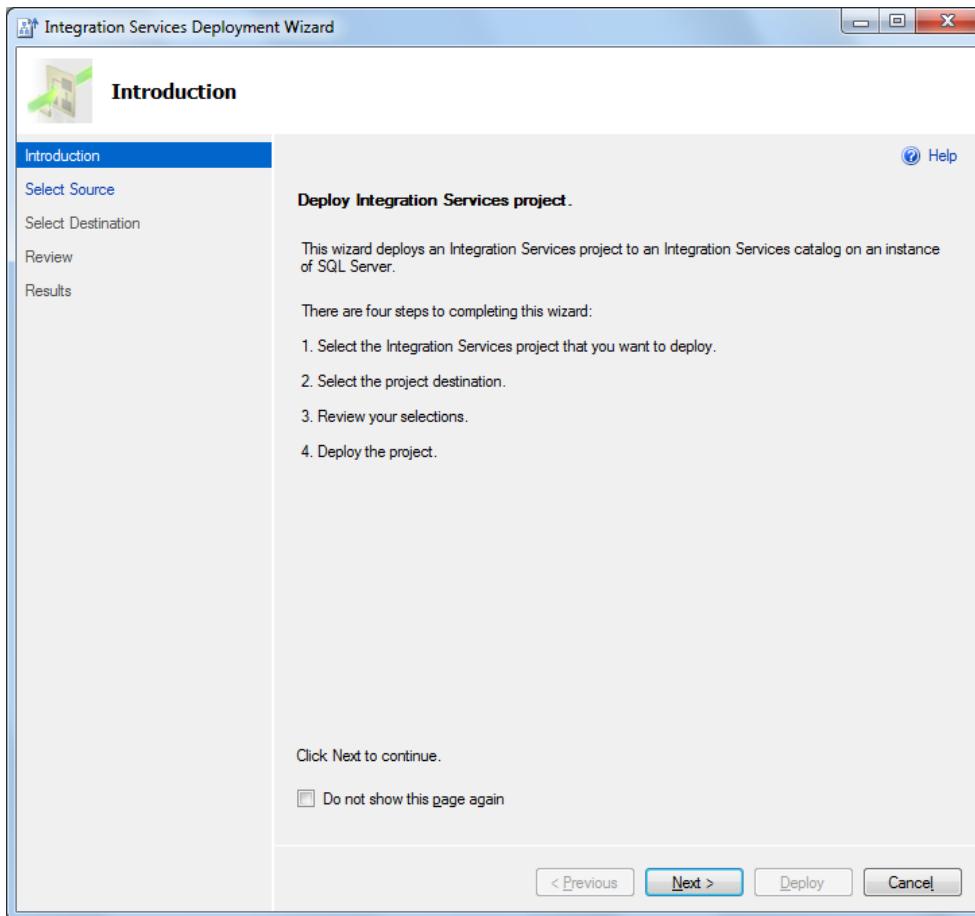


6) Under every newly-created folder we have two sub-folders: 'Projects' and 'Environments'.

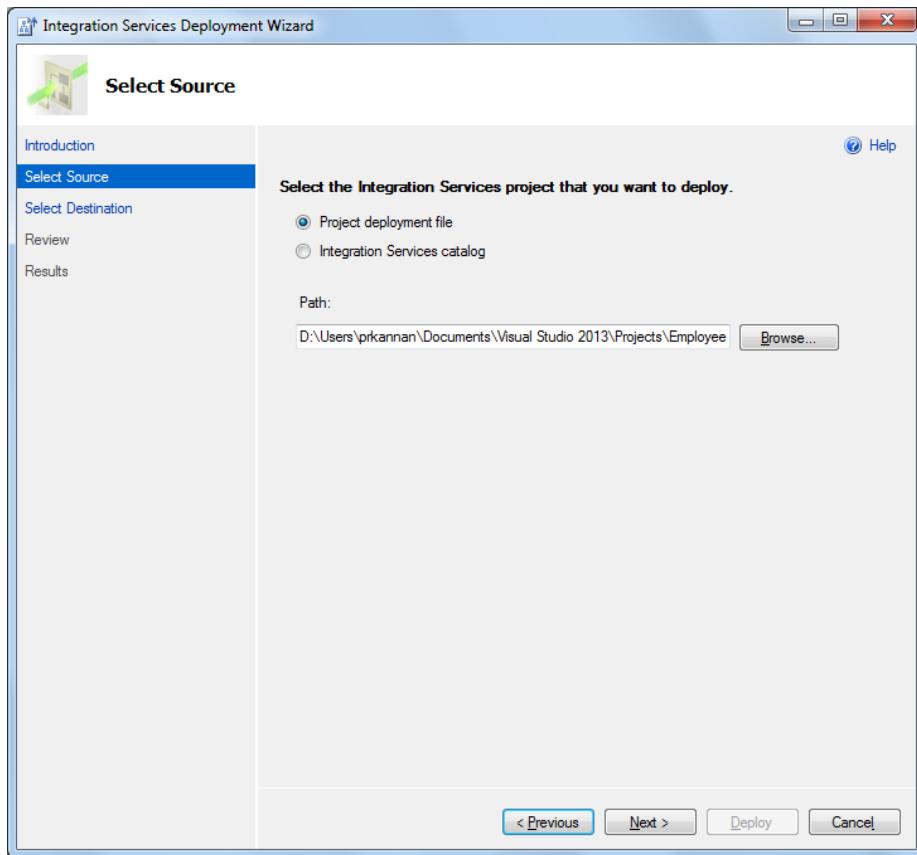


IN SSDT:

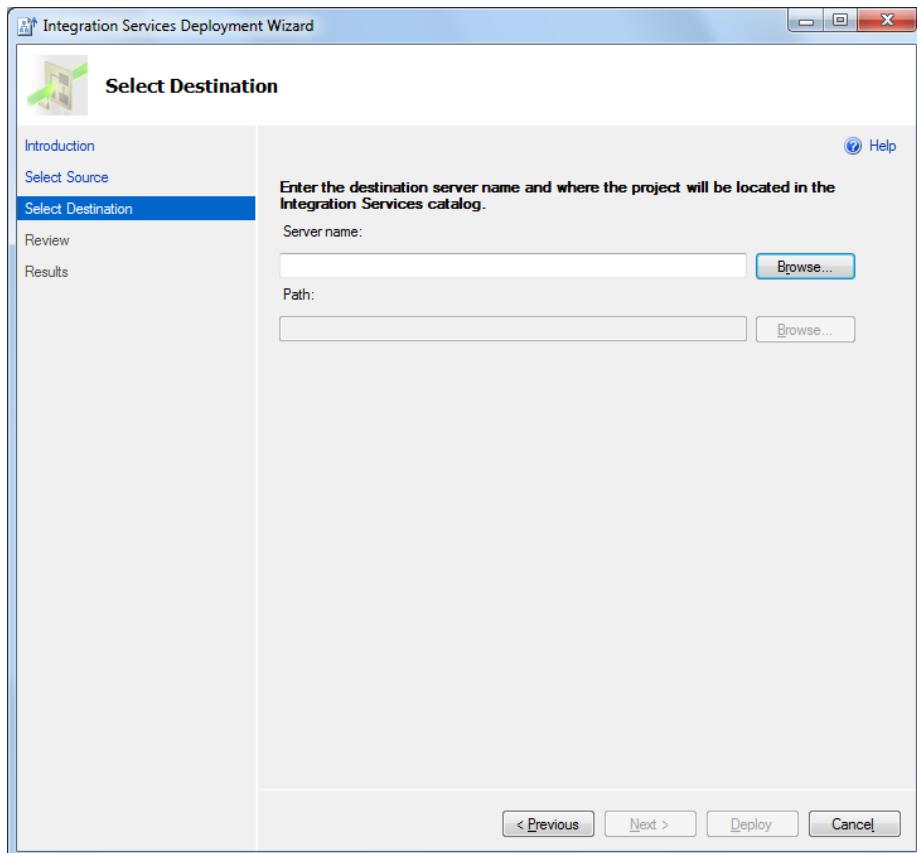
7) Now let's deploy the project to our SQL Server instance. By right-clicking on the project solution in SSDT and clicking Deploy, we get to the following wizard:



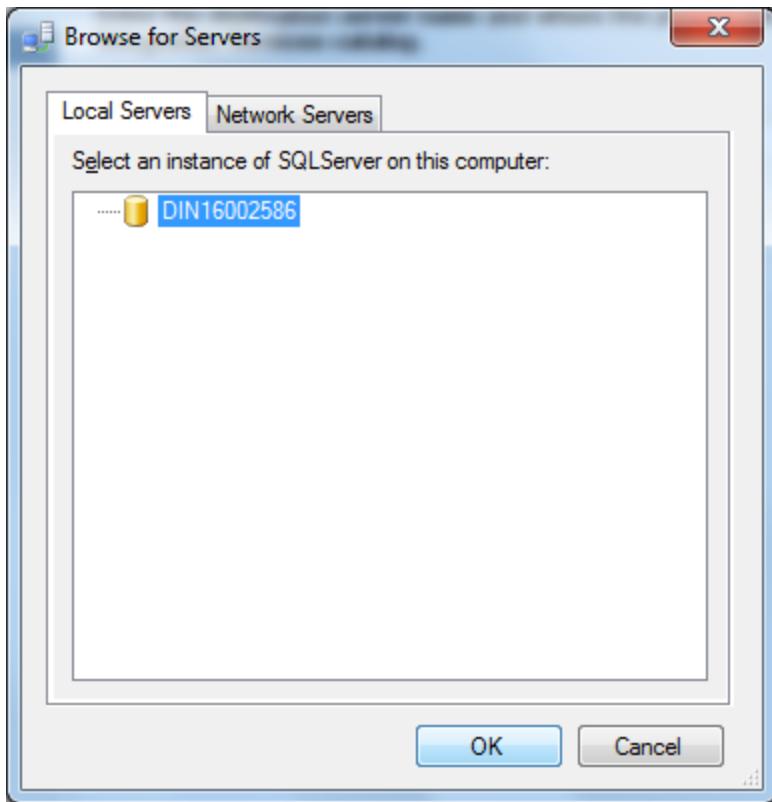
8)Select source page will appears like this:



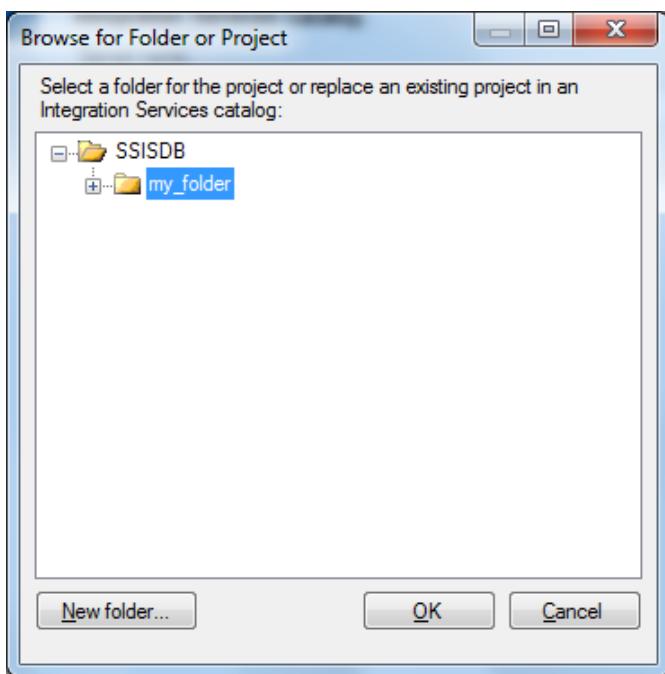
9)Select Destination server looks like the below screenshot:



10) Click browse button to choose your destination server name like below screenshot and click ok.



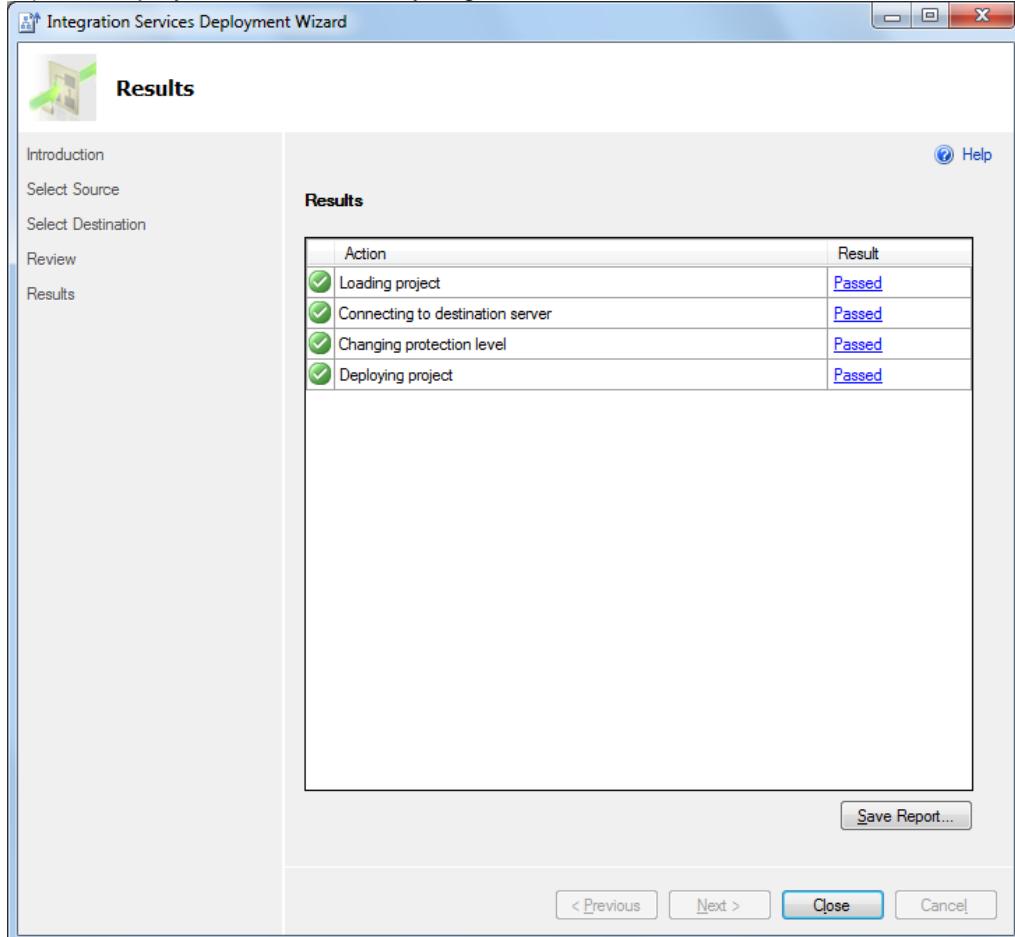
- 11) Now browse path where the project should be deployed as given below.
- 12) You will get the below screen when you click the browse button.



13)Choose the folder which you have created under Integration services catalog and click ok and give next.

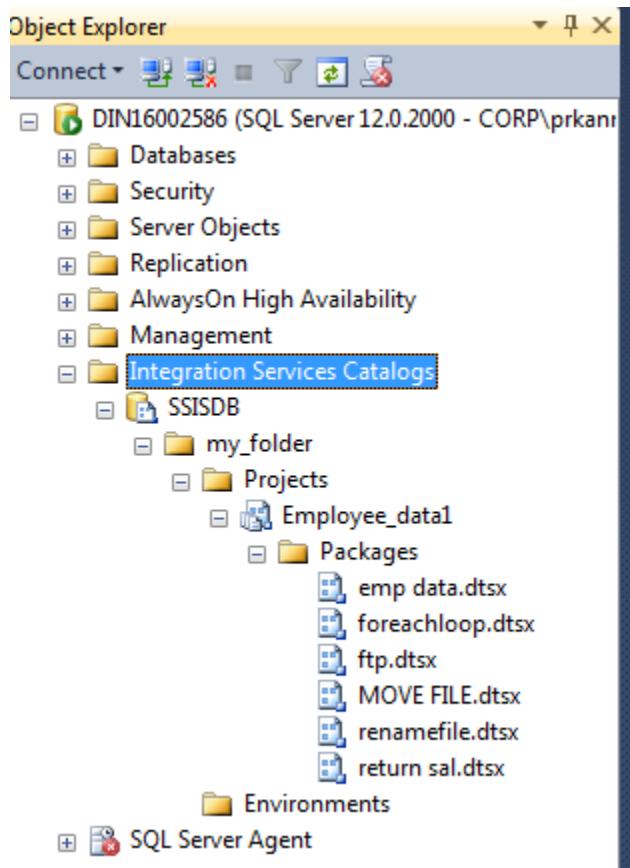
14)Check once for your source and destination and deploy.

15)If the deployment is successful you get like the below screenshot.



16)If you need, you can save the report.

17)The deployed project now looks like this:

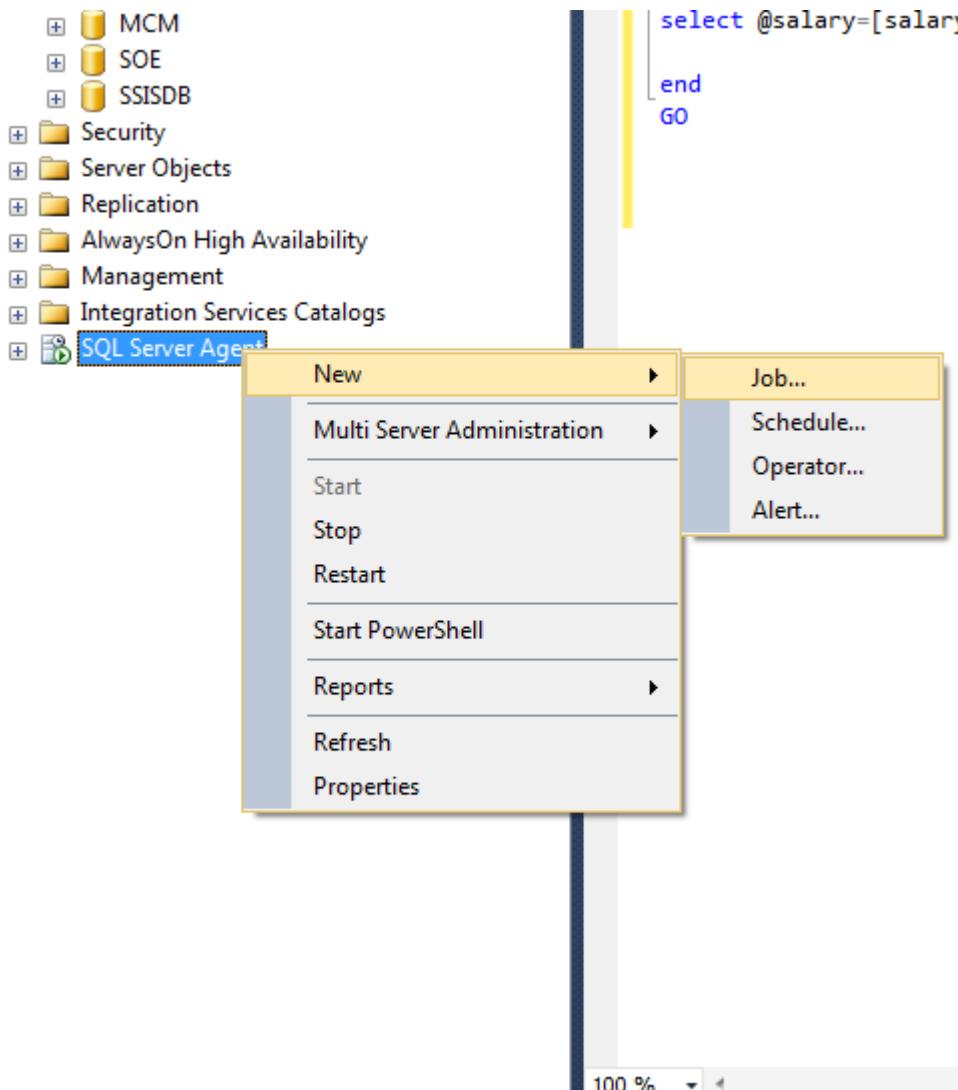


18)From this point on, we can execute the deployed package by scheduling an SQL Server Agent Job and using the package in it.

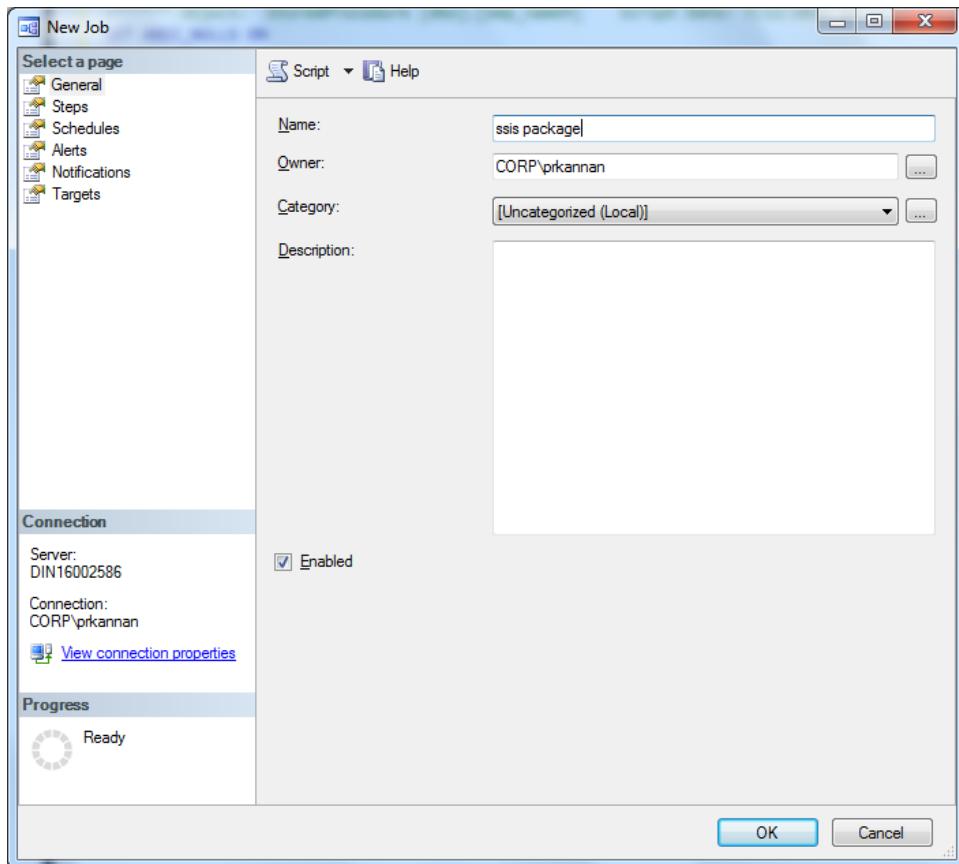
Lab21-Package Execution using SQL Server Agent

Objective	Package execution from SQL Server Agent
Lab Setup	Login details for connecting to the database

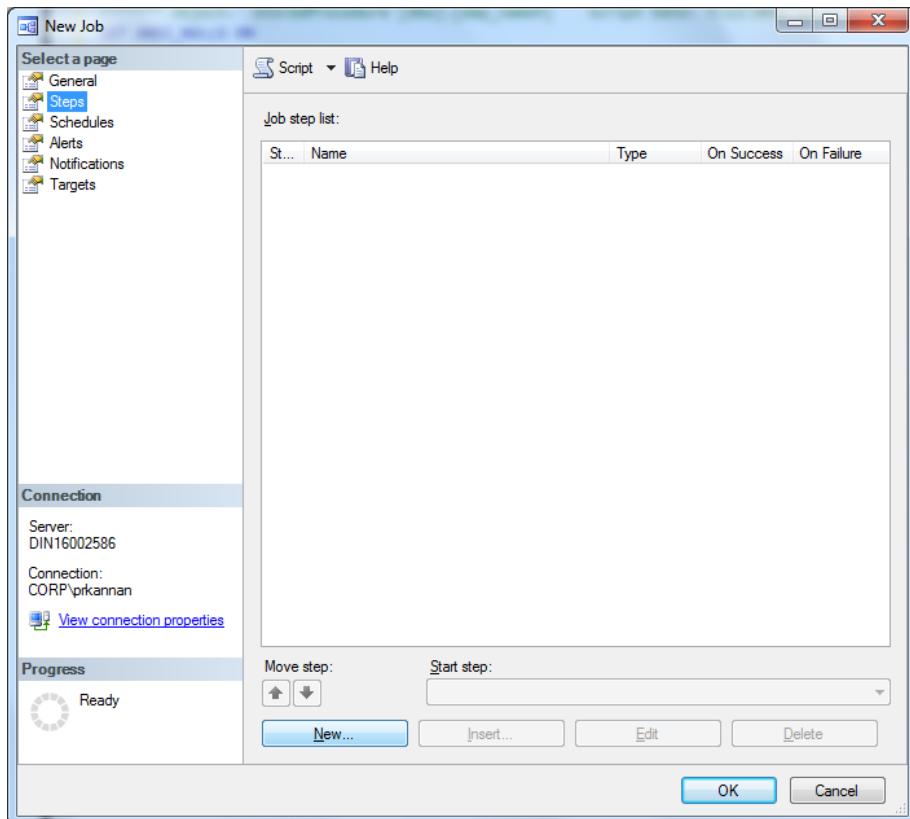
1)To execute a package from SSMS , right click on the SQL Server Agent which will be under your server as per below.



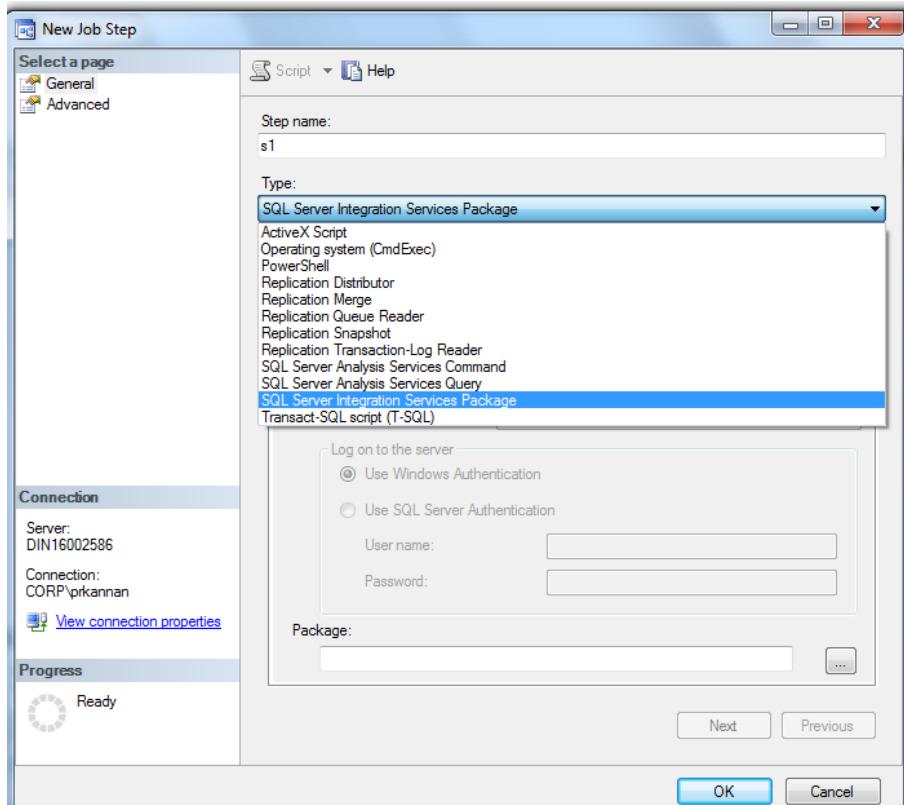
2)Click on new job you will get the below screen. Give name for your job.



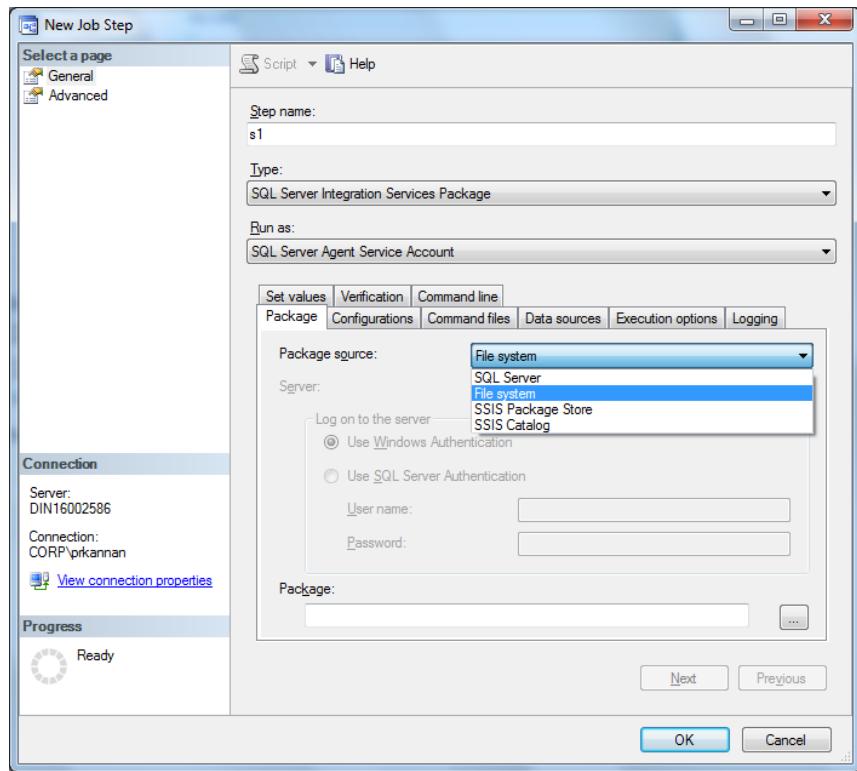
3) Click on the "Steps" under Select a page. You will get a screen like below where you should choose the **New...** option.



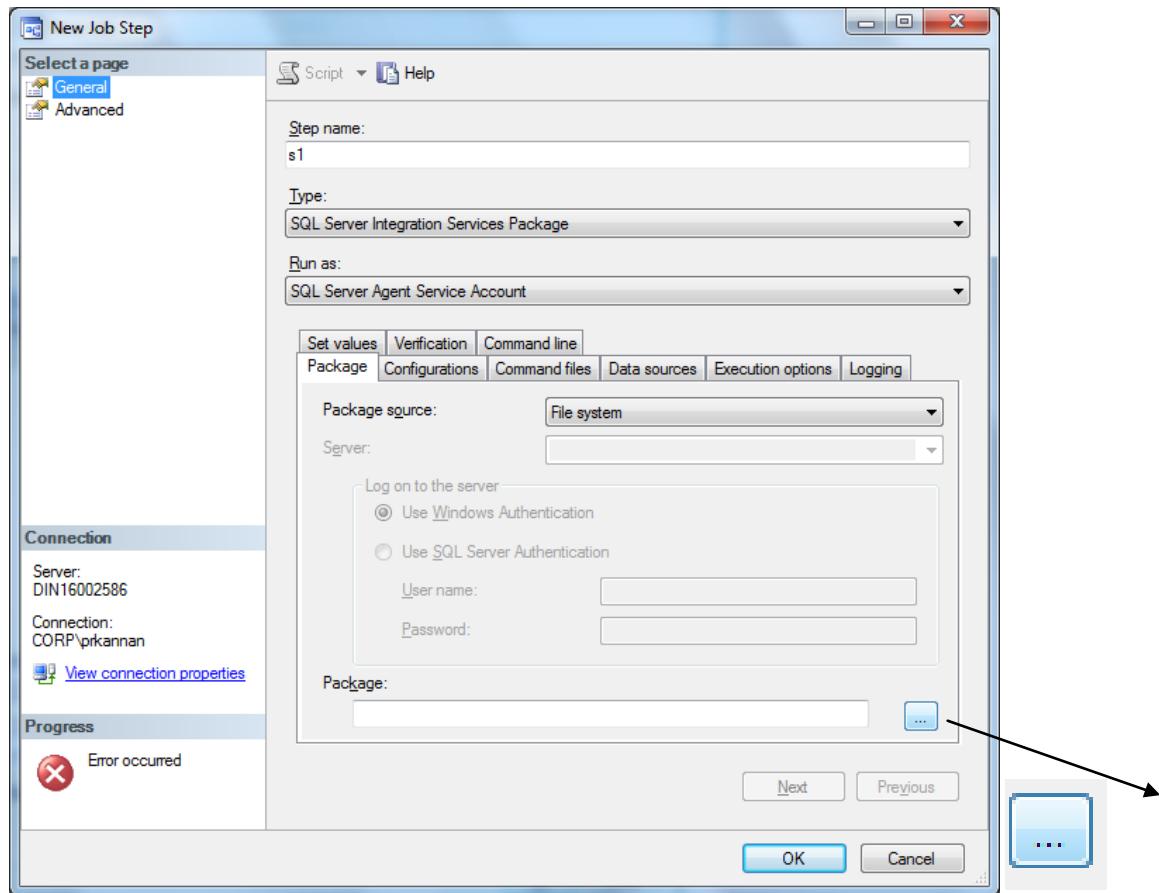
- 4) After selecting new , you will be prompted to this page. Give a name to your step and choose SQL Server Integration Services under type.

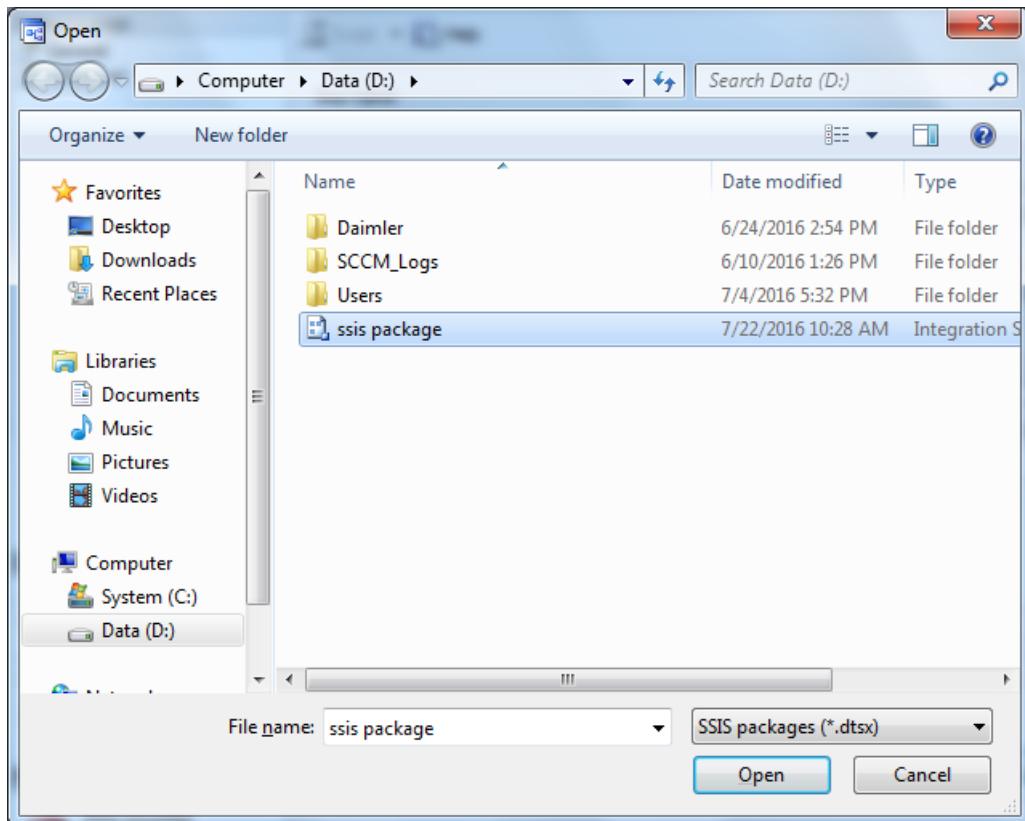


5)Select the package source as File System.

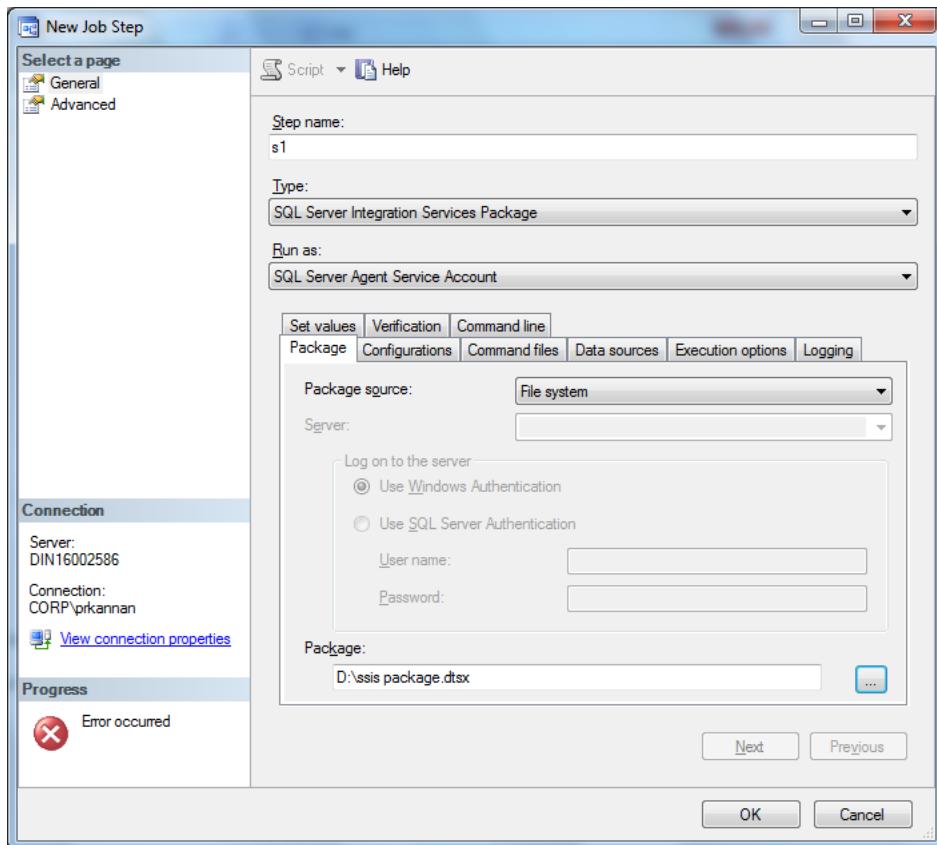


6)Select the package which you want to execute using the browse button as per below.



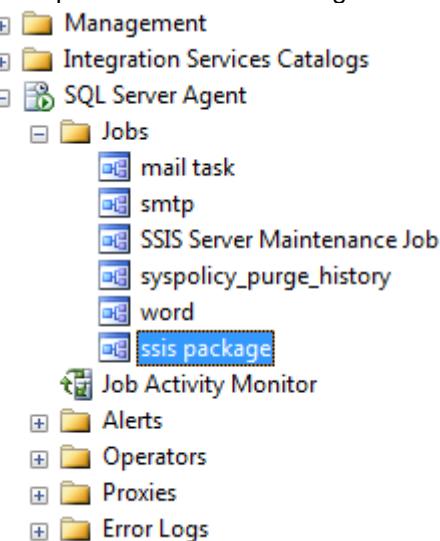


7)After selecting the package click open.

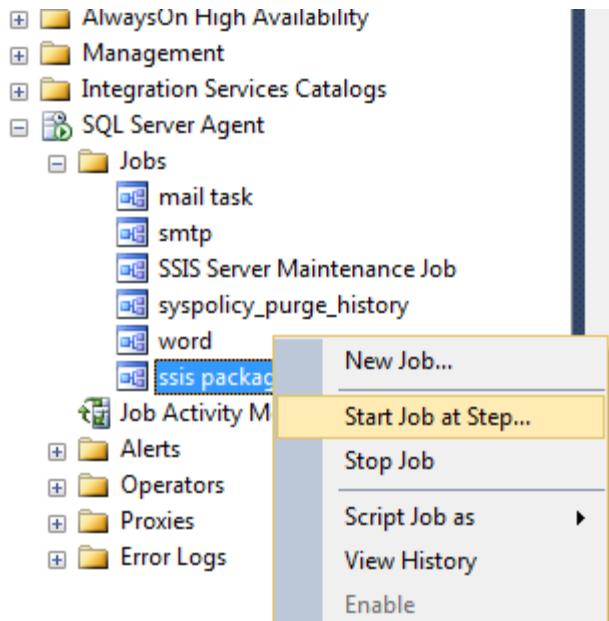


8) Then click ok in the new job step page and new job page.

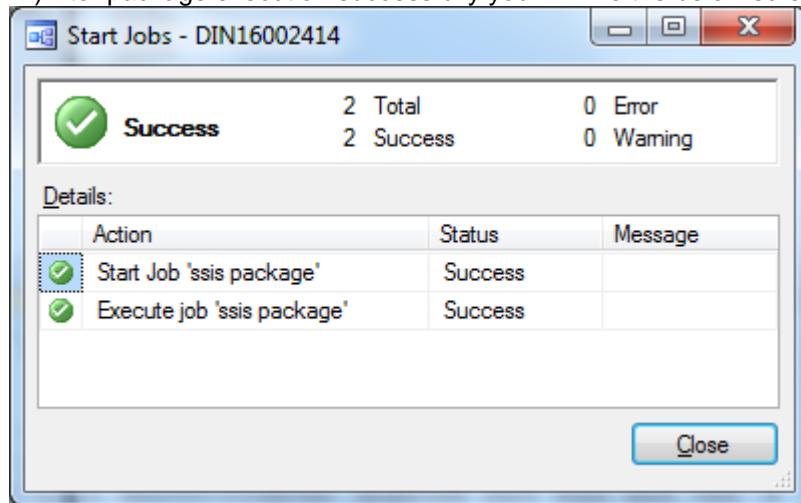
9) Now expand the SQL Server Agent and now you can see your job under server agent.



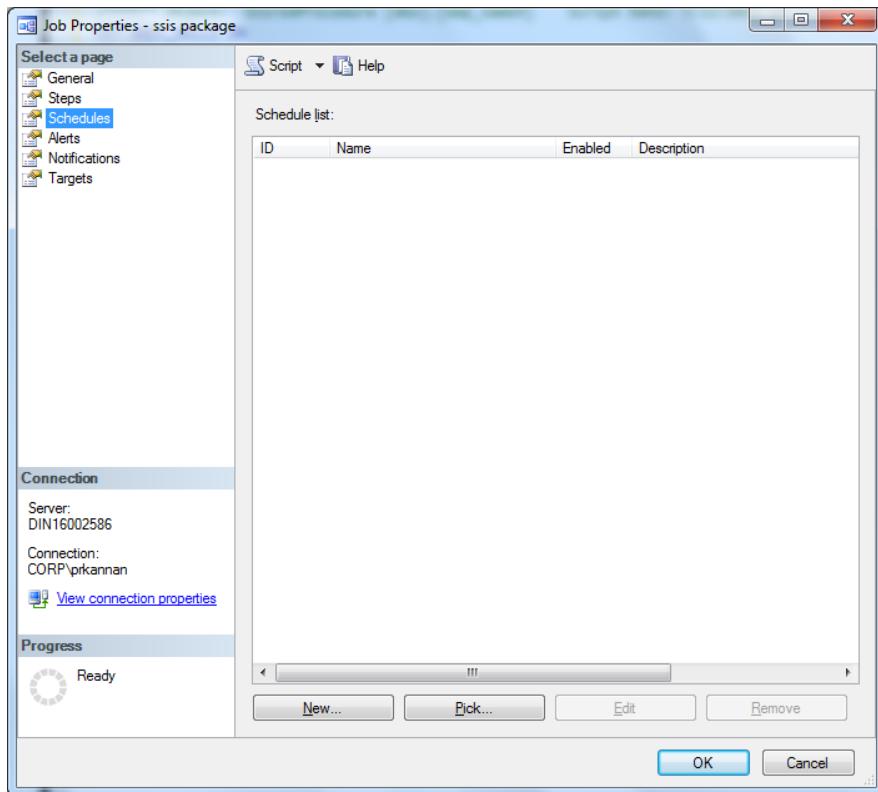
10) Now right click on the package on click Start job at Step.



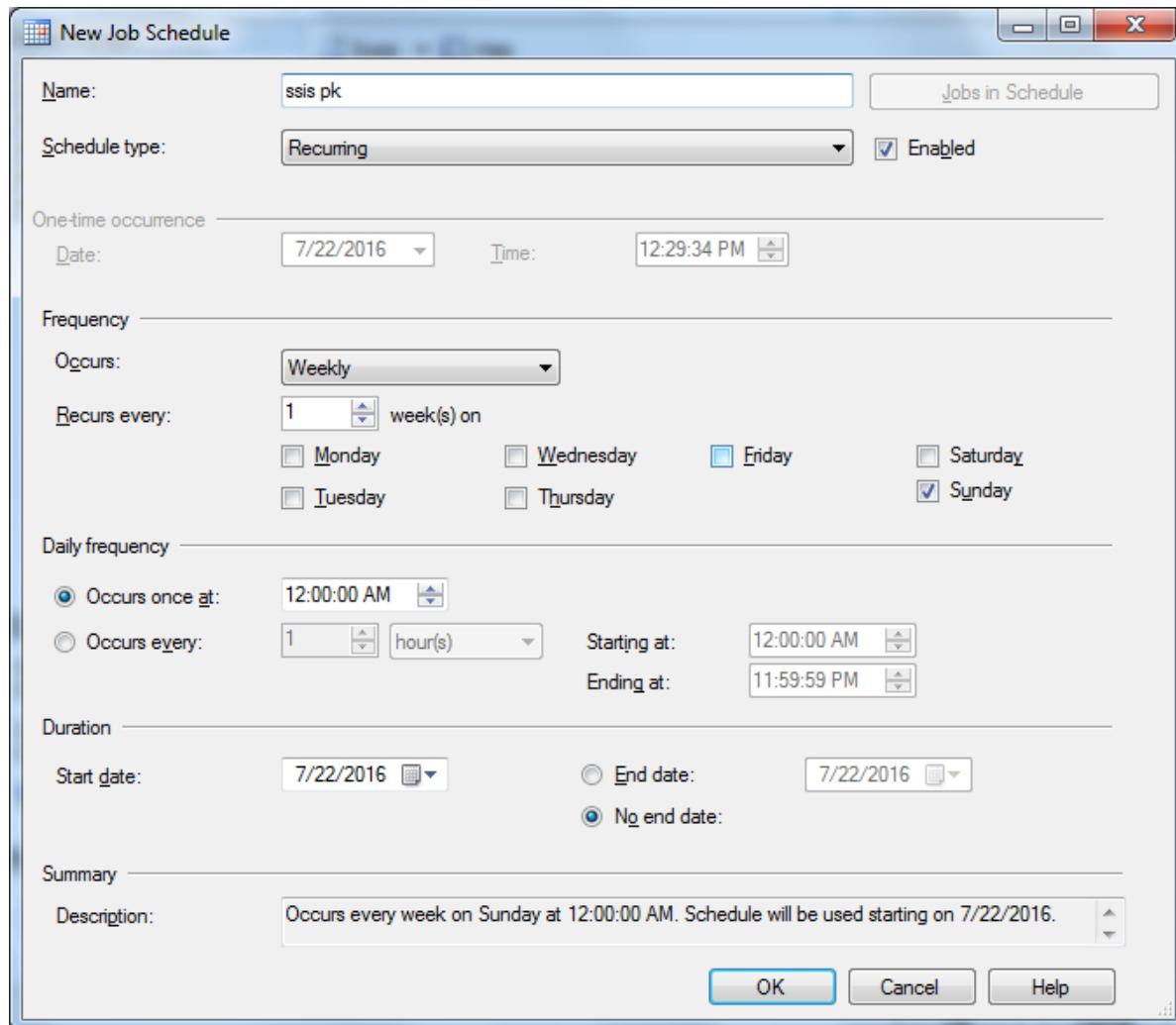
11) After package execution successfully you will like the below screenshot.



12) After executing your package by sql server agent you can schedule your package to run daily on at a frequency you wish. To schedule the job ,double click the package under sql server agent you will get the below screen.



13) Click new at present at the bottom of the screen. You will get the below screen.



14)Here you can specify the schedule type, frequency and daily frequency for executing your job.