**SQL Server Integration Services (SSIS) Tasks (material taken from [1])**

**Data Flow**

The SSIS Data Flow is implemented as a logical pipeline, where data flows from one or more sources, through needed transformations, cleansing, and reshaping and then transferring data into one or more destinations. The Data Flow does its work primarily in memory, which gives SSIS its strength, allowing the Data Flow to perform faster than any ELT type environment (in most cases) where the data is first loaded into a staging environment and then cleansed with a SQL statement.

The Data Flow is made up of three components that are discussed in this chapter: sources, transformations (also known as transforms), and destinations. These three components make up the fundamentals of ETL. Sources extract data out of flat files, OLE DB databases, and other locations; transformations process the data once it has been pulled out; and destinations write the data to its final location.

**Data Viewers**

Data viewers are a very important feature in SSIS for debugging your Data Flow pipeline. They enable you to view data at points in time at runtime. If you place a data viewer before and after the Aggregate Transformation, for example, you can see the data flowing into the transformation at runtime and what it looks like after the transformation happens. Once you deploy your package and run it on the server as a job or with the service, the data viewers do not show because they are only a debug feature within SQL Server Data Tools (SSDT). To place a data viewer in your pipeline, right-click one of the paths (red or blue arrows leaving a transformation or source) and select Enable Data Viewer.

**OLE DB Source**

The OLE DB Source is the most common type of source, and it can point to any OLE DB–compliant Data Source such as SQL Server, Oracle, or DB2. Let us do an example that an OLE DB Source can be configured.

1. Drag and drop the Data Flow Task in the Control Flow designer view

2. Drag and drop the OLE DB Source on the Data Flow designer view

3. Double click the OLE DB Source to open the configuration editor

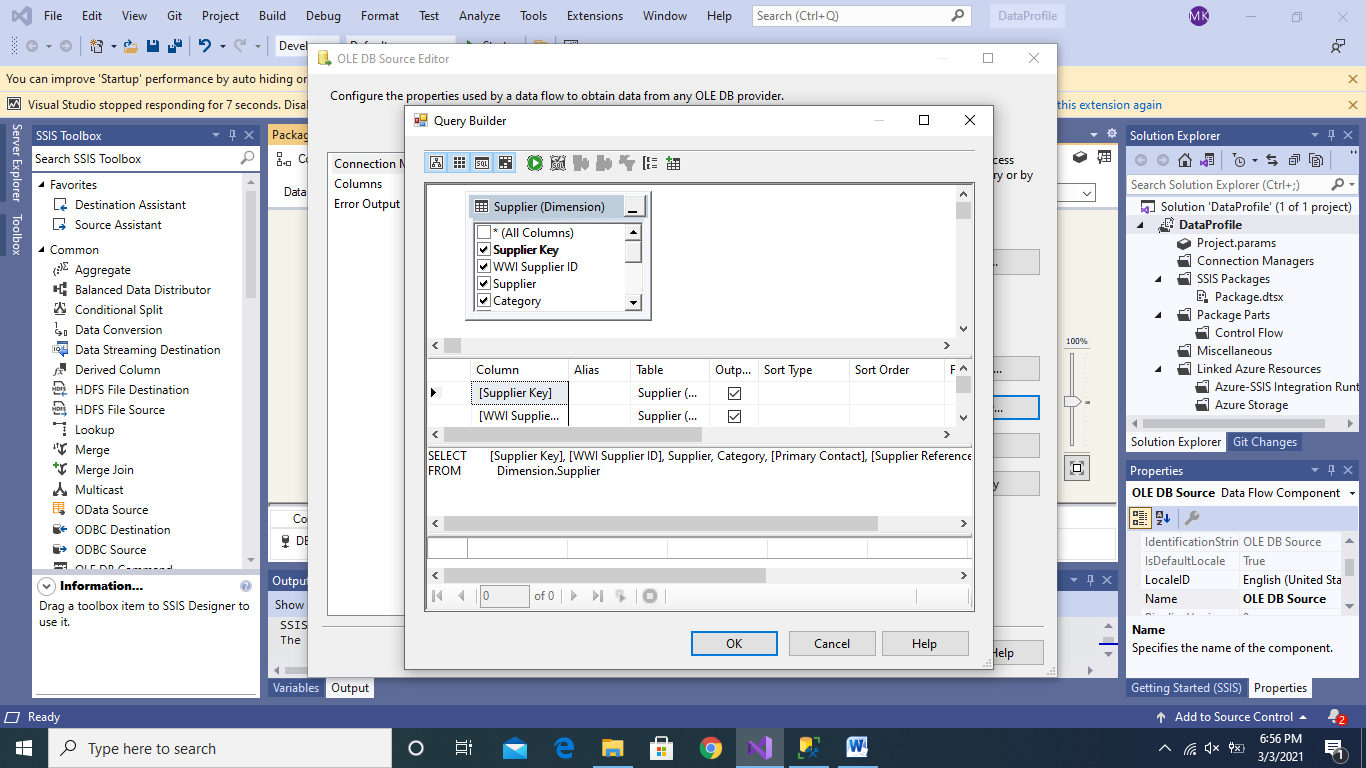
4. Make a new connection with one of your databases hosted on local SQL Server

5. Click the Build Query button to open the query builder

6. Click the Add Table button on the Query Builder dialog

7. Select your desired table or tables from the pop up menu

8. In my case I have selected only one table and after that I have selected few columns from the selected table as shown in the following figure



9. The Query Builder helps in writing SQL Query as show in the above figure

10. Next, click Ok

11. Next, select Columns tab in the configuration editor

12. Here you can select Columns that you want to send as output for the purpose of transformation or for a destination

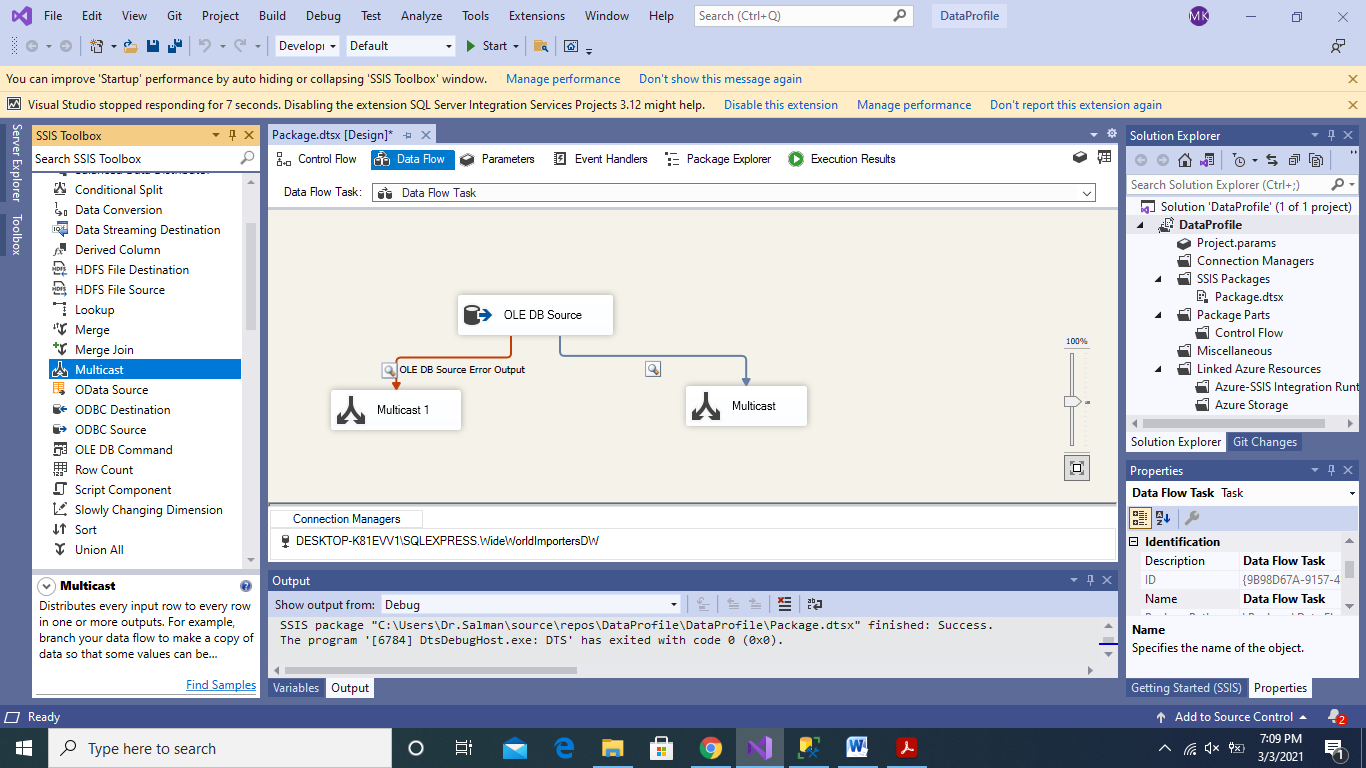
13. You can also change the name of the output column

14. Next, select the Error Output tab. Here you can select if any error happens in the data type or field length truncation then what to do. There are three options here, i.e., fail the whole process, ignore the error and redirect the problematic row to a particular destination

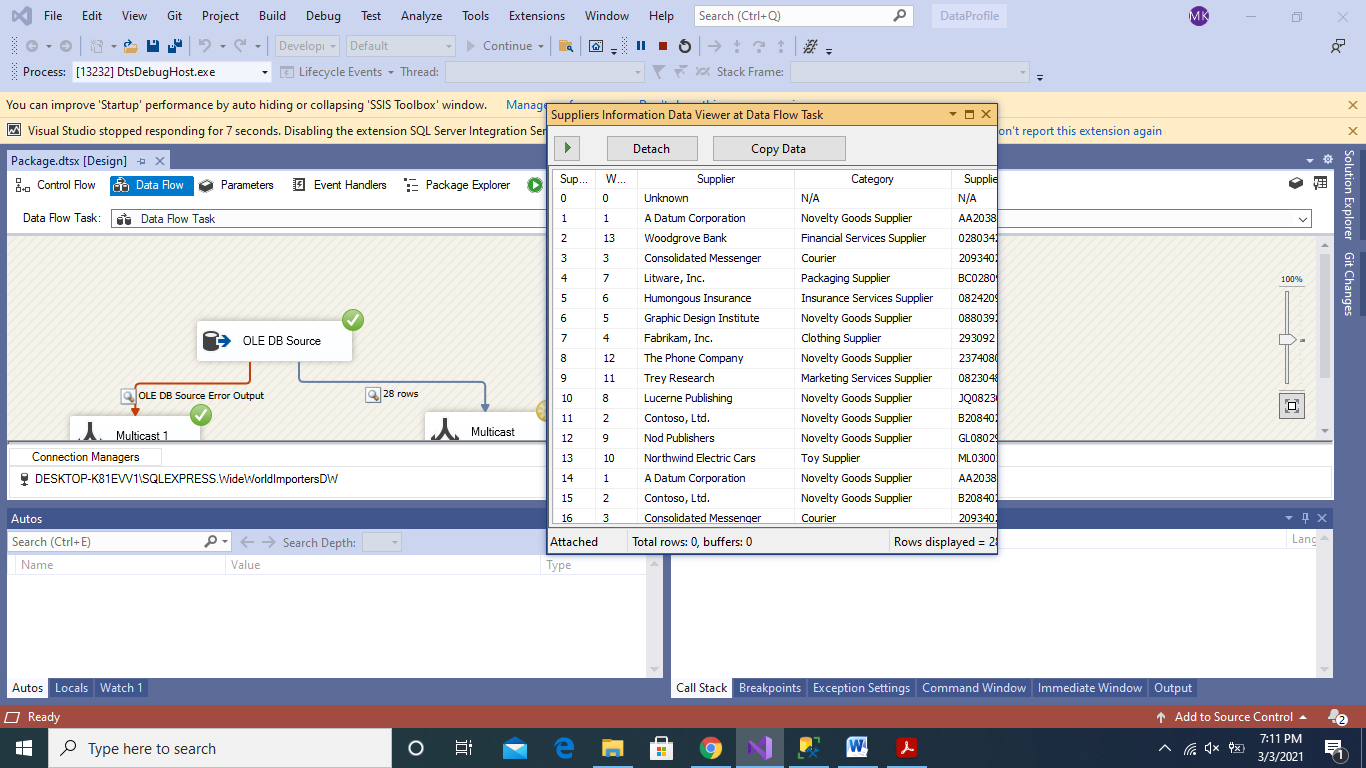
15. Next, drag and drop two MultiCast transformation. Link one MultiCat with blue arrow of OLE DB Source and the other with red arrow. The red arrow is for if any error happens as mentioned in point no. 14 then insert the faulty row in the destination pointed by red arrow. However, you need to make appropriate selection for the Error Output as mentioned in point no. 14 as above. The purpose of the MultiCast transformation is just to debug that the data is being pushed in the pipeline

16. Next right click the red and blue arrow and enable data viewer so that we can debug that whether the data is being pushed properly from the source

17. The final OLE DB source and MultiCast will look like the figure below



18. Now run the package, if there are no errors then you will see that data viewer will pop-up and you will be able to view data as shown in figure below



19. Click the green > button on the data viewer to see next batches of data if the data is more than 10,000 rows.

20. Since there are no error in the output the data viewer of red arrow will not show an data.

**Excel Source**

Repeat the above mentioned steps for the Excel Source. An Excel file with 50000 records and with the name Orders\_Excel.xsl has been uploaded on our Google Classroom. Please note that with this source we cannot change the data types of columns therefore we need Data Conversion transformation to transform the data types of Excel columns into appropriate data types for the destination

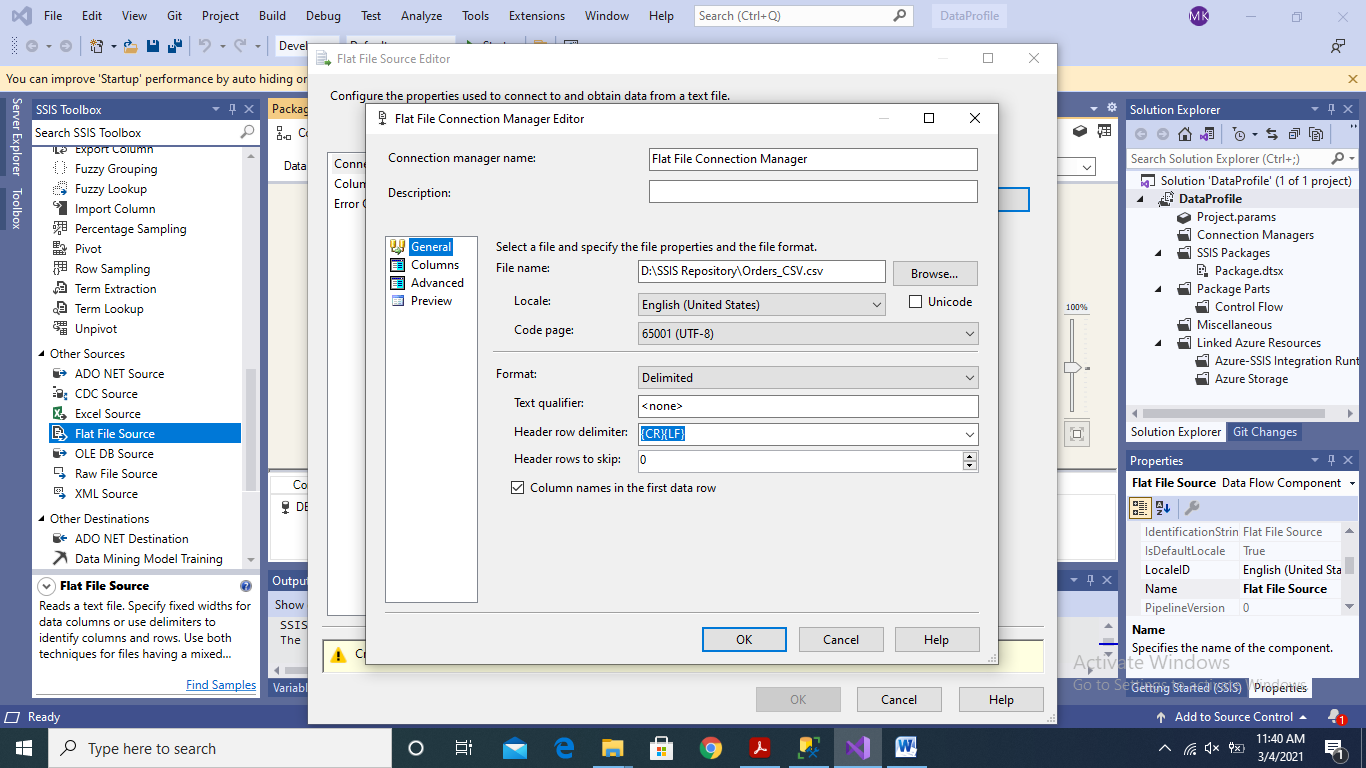
**Flat File Source**

In this task, we will use Orders.csv file to understand different options for the configuration manager

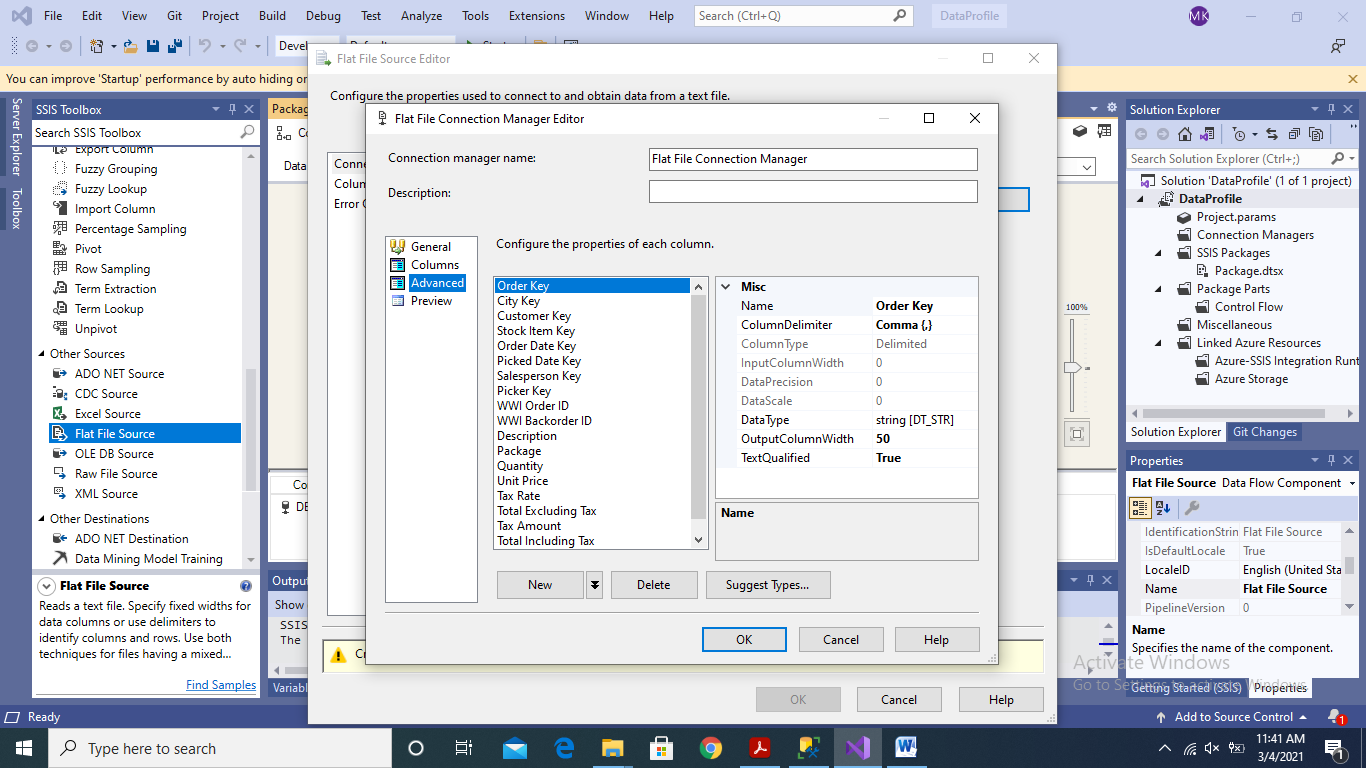
1. Drag and drop the Flat File Source in Data Flow designer view

2. Make a connection with Orders\_CSV.csv file. This file has been uploaded on our Google Classroom

3. Make the configuration as shown in the below figure. This is actually the default options



4. Click, on the Advanced Tab on the configuration editor. The following view will open in the dialogue box.



5. As it can be observed from the Advanced Tab that here in this dialogue, I can change the name of the columns and I can also change the data type of each column. By default, we can observe that majority of columns have been given default data type as string in SSIS. You can change these data types and their field lengths as per your requirements of the destination.

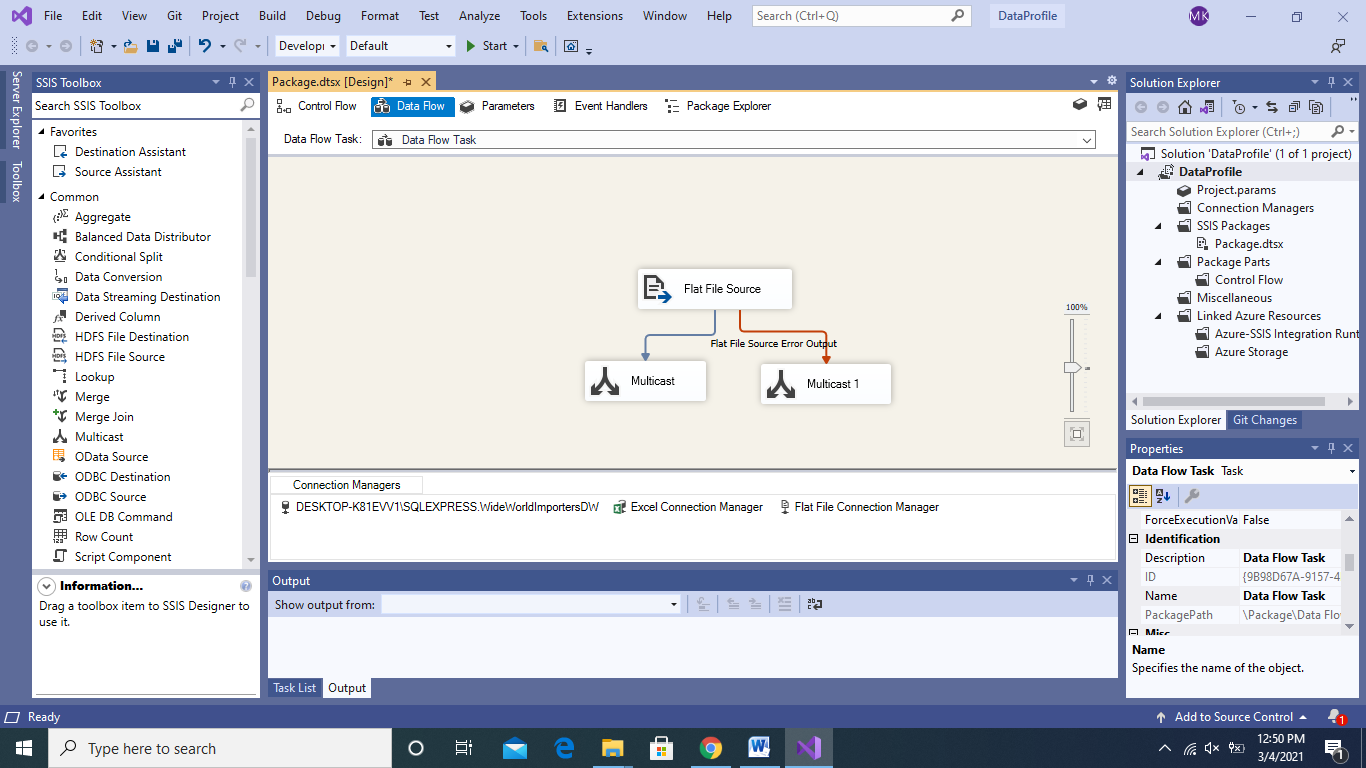
6. Here in this dialogue, you can also click on “Suggest Types” where SSIS selects data types for each column. Sometimes it proposes correct data types but sometimes you need to decide the data types by yourself. Therefore before sending data for the Data Flow, we must need to audit the data types of each column.

7. In my case, I have selected four-byte unsigned integer type for the “Key” fields. For the Order Date Key and Picked Date Key columns, I selected date data type in SSIS. For the Description column, I have increased the field length to 100 characters. Since this field does not contain Unicode characters therefore, I have selected only simple String data type.

8. For the rest of the columns, please select appropriate data type

9. As in previous tasks, drag and drop Multicast transformations to debug normal data flow and error data flow. For this purpose you also need to enable “Data Viewer”.

10. The final solution will look like the figure below



11. Run the package, if there is no error then you will see the data in the Data Viewer on the normal data flow. If there are errors and you have mapped the error to the Error Data flow. Then you will be able to see the information of the records that have errors and error information on the error data flow with the help of Data Viewer

References:

[1] Knight, B., Knight, D., Moss, J. M., Davis, M., and Rock, C.: “Professional Microsoft SQL Server 2014 Integration Services”. Publisher: Wrox