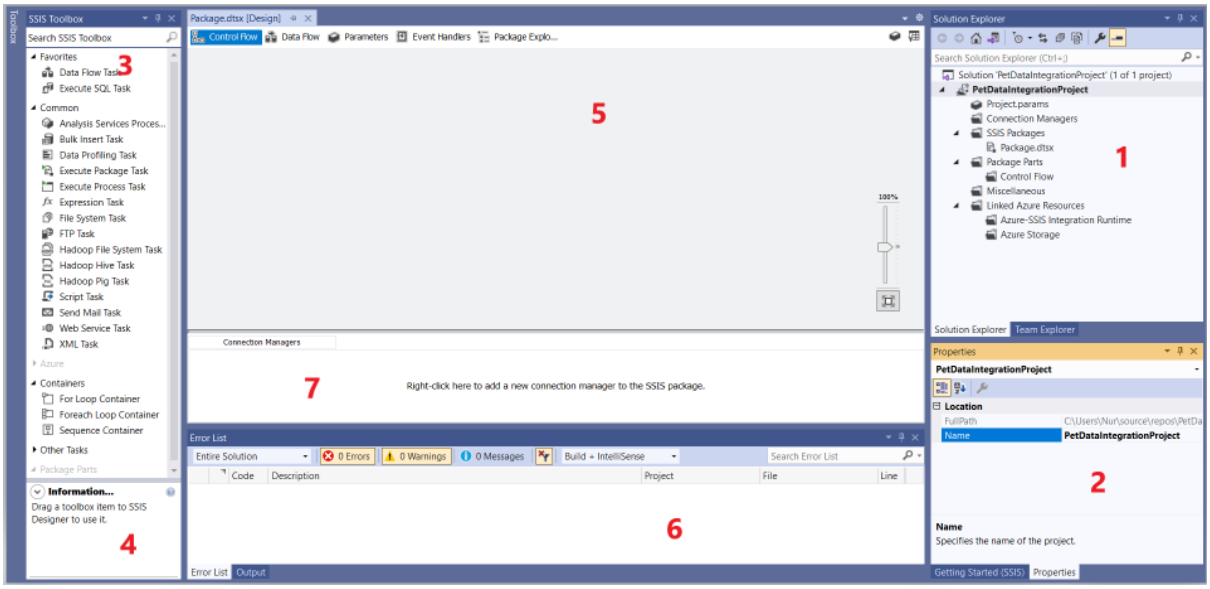
Lab Week 5

Integration Service

The Microsoft SQL Server Integration Services (**SSIS**) has many built-in tasks and transformations to solve complex business problems by building high-performance data integration packages. The SSIS is a business intelligence tool that provides data transformation solutions for various organizations. One can use SQL SSIS for updating data warehouses, data mining, downloading or copying files, extract and transfer data from XML to SQL, etc. In other word, SSIS can be used to extract data from a wide variety of sources such as **Excel Files, Flat Files, XML Files, Relational databases**. Furthermore, transform (slice and dice) them as per your requirements and finally load the data into the destination. To develop or create an SSIS package, you need SQL Data Tools.

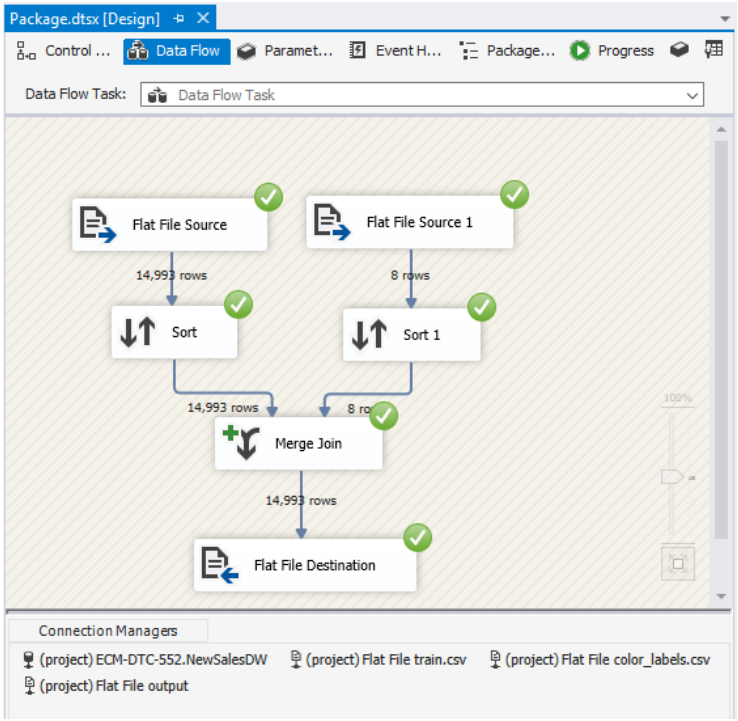
In this Lab, we will demonstrate how SSDT (SQL Server Data Tools) can be used to process data by building a Microsoft Integration Services Project. Below, we show the features and descriptions of SSIS work environment.



1. **Solution Explorer:** This is a combination of project level connection managers, actual packages, and project parameters.
2. **Properties:** Contains the properties of a project. We can change the properties of a task from this pane.
3. **Toolbox:** SSIS Toolbox provides a lot of built-in tasks, containers, transformations, sources, destinations, and administrative tasks to solve complex business problems. Use these graphical SSIS tools by drag and drop those tasks in the work environment. This means, we no need to write a single line of code to perform most of the operations.
4. **Information:** Shows the information about the toolbox items
5. **Package:** Design SSIS package
6. **Message:** Shows output and error messages
7. **Connection Managers:** This window is to create a package level connection managers

Task 1: Using a SSIS Project solution to merge join two flat files

SSIS provides a very useful GUI for a multitude of data manipulation tools, while provide really nice visualisation to document the data transformation process. For example, the screenshot below is the end result of joining two files in SSIS, which makes documentation and record keeping of the data transformation really natural. The whole process does seem to be overkill for this simple merging exercise, but for more complex data wrangling, the benefits certainly outweigh the learning curve.



1.1 Preparation - Resources needed for this lab include:

Download the pet-data.zip from blackboard.

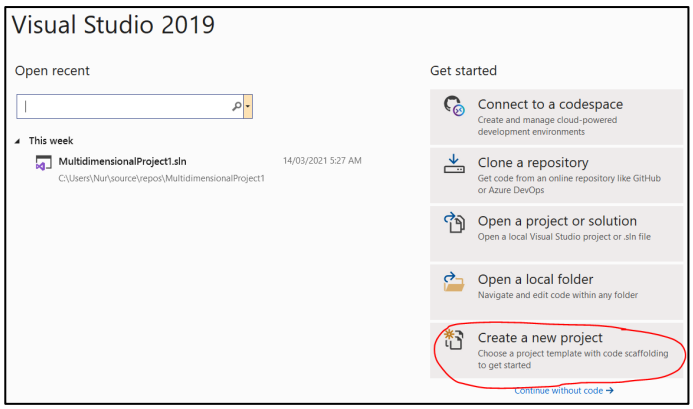
- train.csv
- color_labels.csv
- train_processed.csv (a new file)

  Create a new file named train_processed.csv and copy the first line of train.csv file, which is a list of comma separated column names, and insert the new column name Color1_name next to Color1.

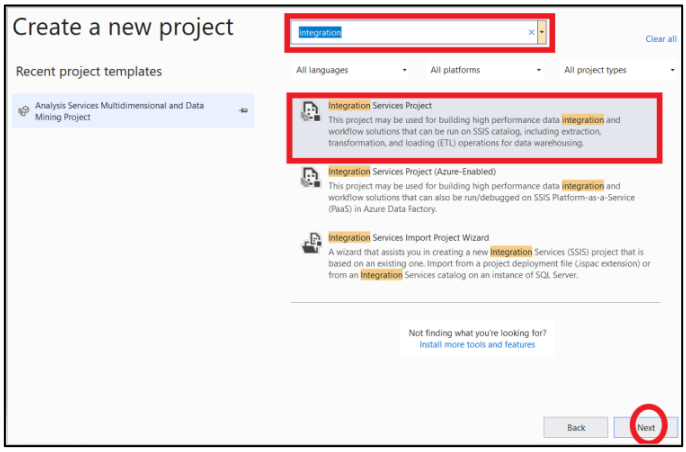
  "Type","Name","Age","Breed1","Breed2","Gender","Color1","Color1_Name","Color2","Color3","MaturitySize","FurLength","Vaccinated","Dewormed","Sterilized","Health","Quantity","Fee","State","RescuerID","VideoAmt","Description","PetID","PhotoAmt","AdoptionSpeed"

- Optionally, if you would like to write directly to the database, a new schema table that has all columns of train.csv with an additional column Color1_name.

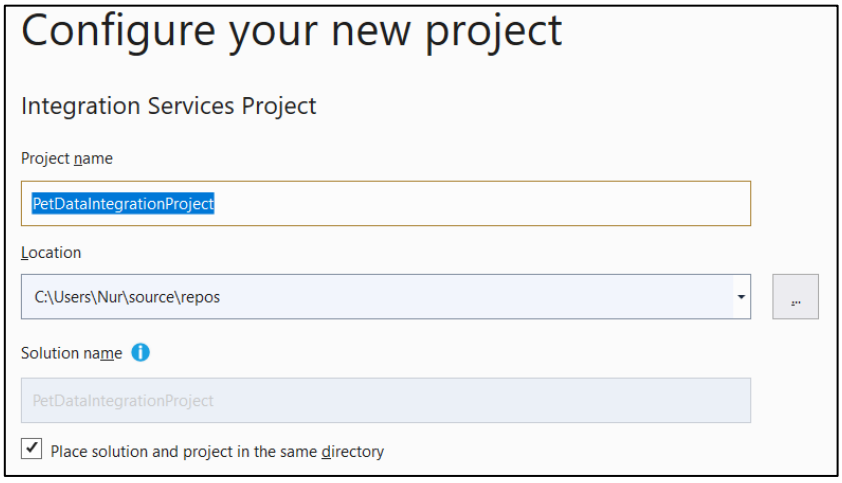
1.2 Open Visual Studio and create a new project of type Integration Services Project.



Search for “Integration” and click on “Integration Service Project”



Name the project as appropriate or keep the default. For example, we use the project name as “PetDataIntegrationProject”.

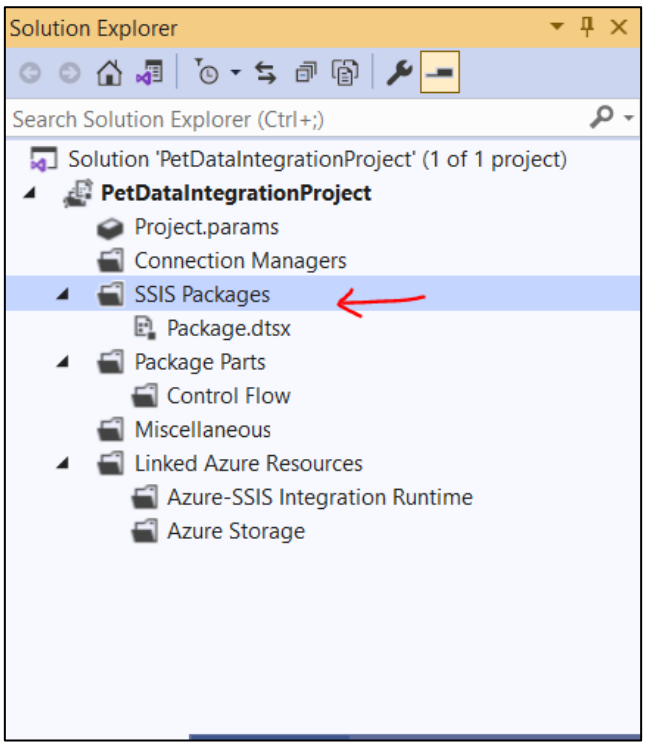


Now, the main window of SSIS will open. The above steps are used to create a new SSIS project.

1.3 Create an SSIS Package

When you create a New Project, SSIS automatically creates a new package. However, you have a choice to create a new package in SSIS under a project.

a. To do so, right-click on the SSIS Packages in Solution Explorer, and select New SSIS package option from the context menu.



b. It will automatically create new .dtsx package. One can rename the package name.

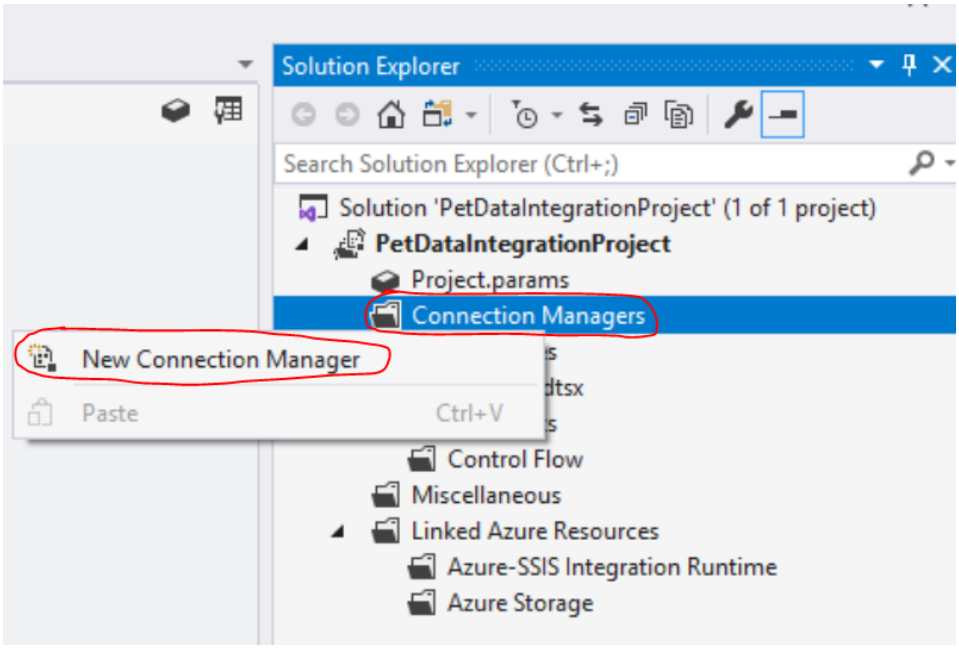
1.4 SSIS Data Flow

There are three types of SSIS data flow components: **Sources**, **Destinations**, and **Transformations**. Each data flow component has an output, and you can use the output to connect with other components.

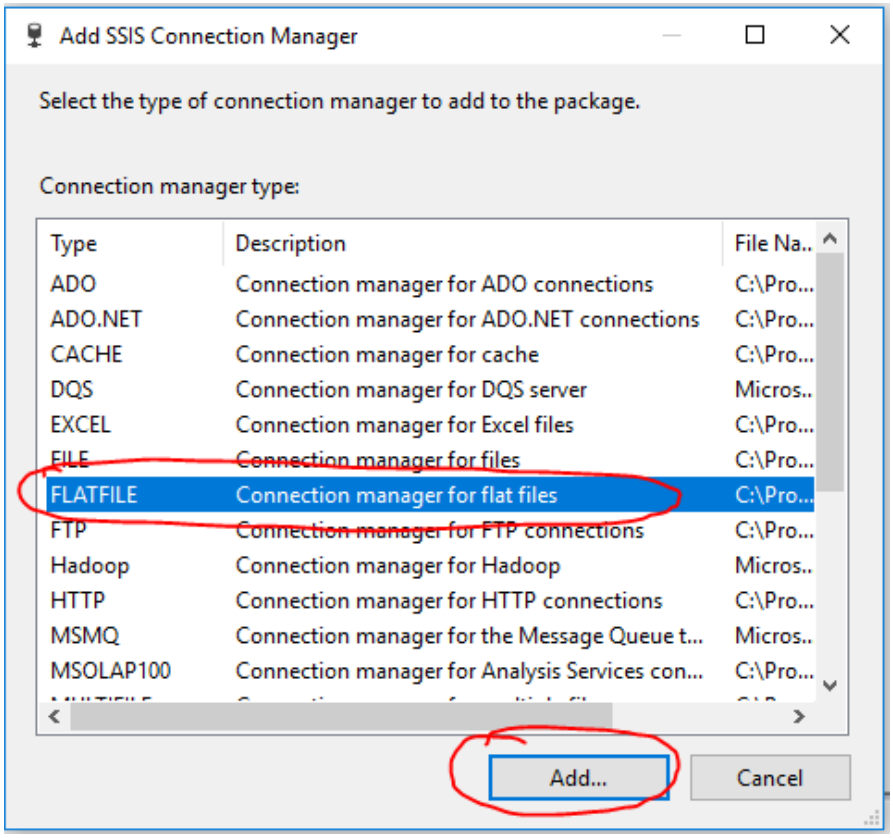
a. SSIS Sources: The main target of the SSIS package is to transfer data from heterogeneous sources to a destination. It means, one need a Source to get the data from and a Destination to load data into it. The sources can be an Excel Source, OLE DB Source, Flat File Source, etc. In this lab, we will use Flat File Source to extract or reads data from text/csv files.
b. SSIS Destination: Destination is used to write data to file format present in the File System. For example, in Flat File destination, the text file can be in fixed width, delimited, ragged right, or fixed width with row delimiter.
c. Transformations: The SSIS transformations are the data flow components that are used to perform aggregations, sorting, merging, modifying, joining, data cleansing, and distributing the data.

1.5 Create Connection Managers

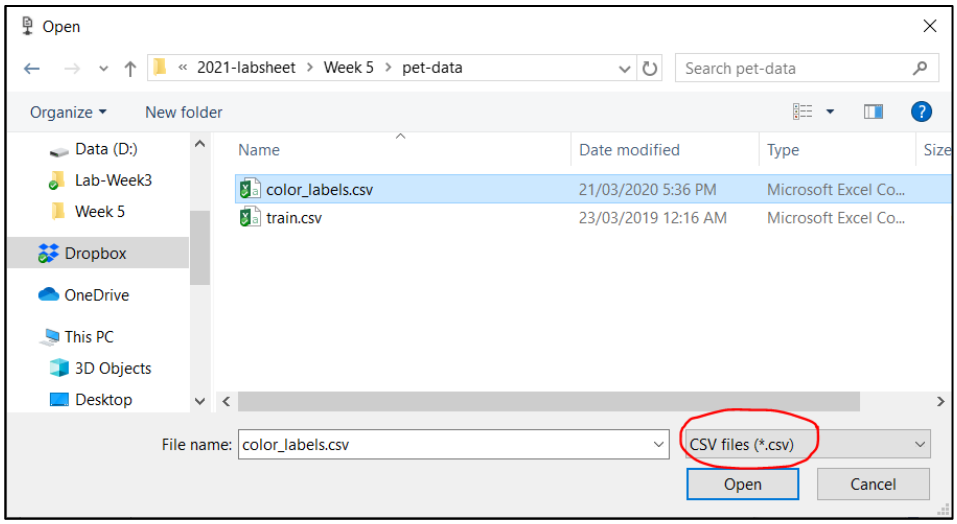
Right click the Connection Managers on the Solution Explorer.

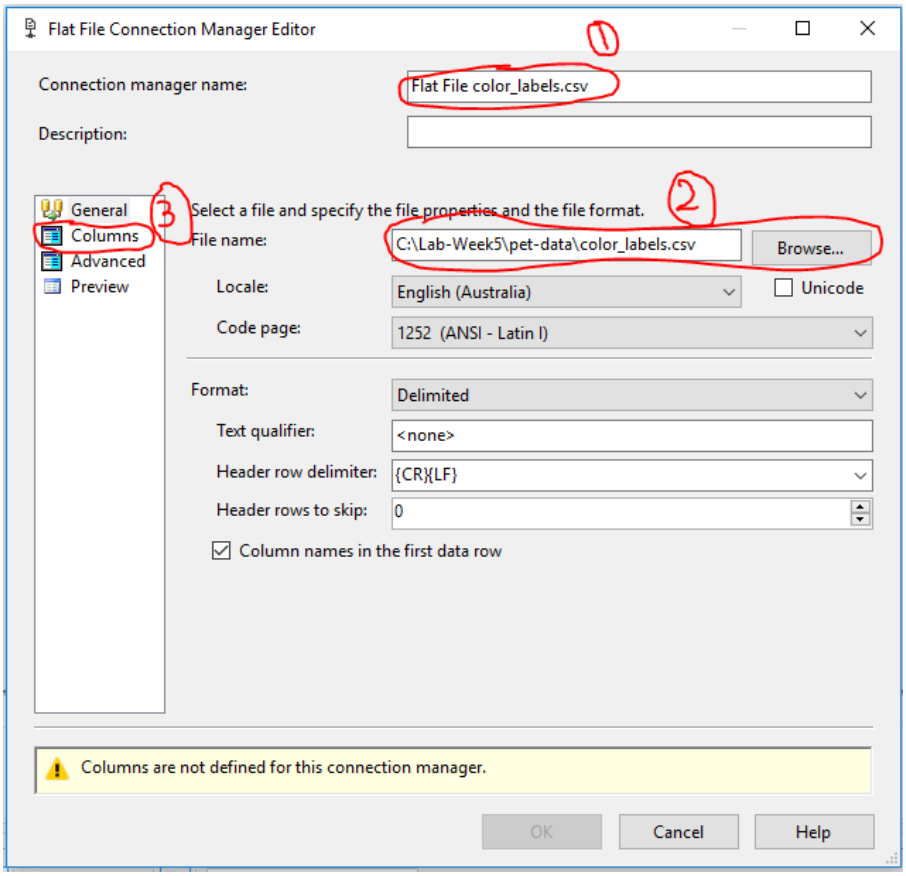


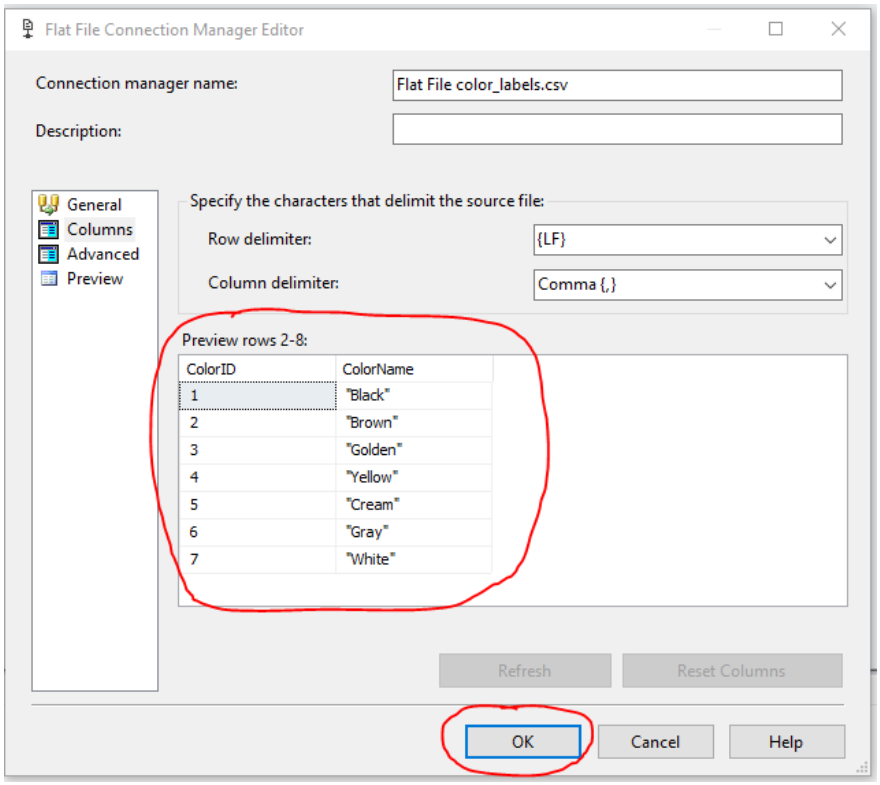
Click on "create new connection manager" will bring a window "Add SSIS Connection Manager". To realise the above merging scenario, we now need to create connection managers for the three flat files, two as inputs to read the data from (train.csv, colors_labels.csv), one as destination output for storing the processed output (train_processed.csv). Alternatively if you have a database table ready then we can send the output to a MS SQL database table.



Create one Flat File Connection Manager for color_labels.csv, note you will need to click the Columns section to verify the columns breakdown is fine. This will also remove the warning message and enable the OK button for it to proceed further. Browse the color_labels.csv file.



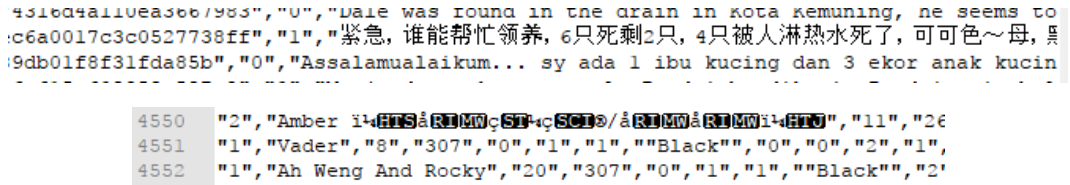




The color_labels.csv file is a very small and simple table, in the sense that it does not contain uni-code and really long text.

If you do the same for train.csv, then there will be a lot of problems because the data contains international characters in both the name and the description column, and the description column can be super long, containing much more characters than the default 50 set by SSIS. In addition, these two text fields also contain commas, which makes plain parsing incapable of separating the columns properly.

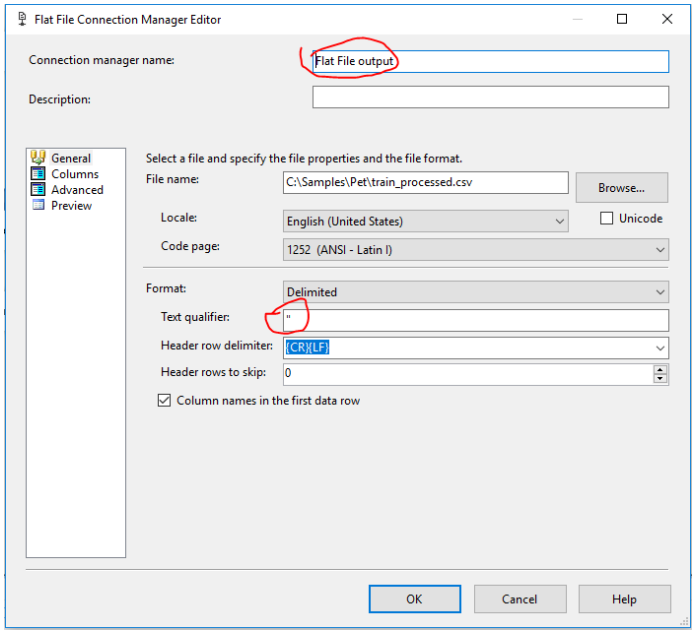
Below a screenshot from a plain text editor to show that there are Unicode text in the train.csv file.



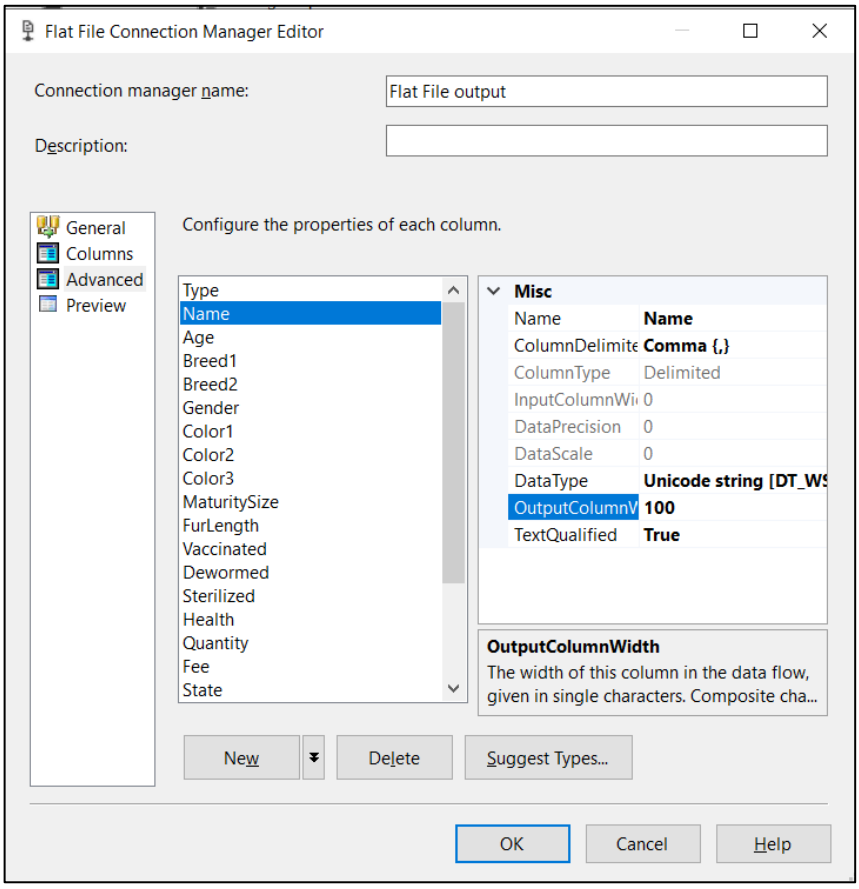
Use the default settings for train.csv and click on Preview, take a look at Row 4, and see if the PetID has been extracted properly.

To address all these problems, we need to

- Indicate the Text qualifier is a double quote “, for both train.csv and train_processed.csv
- For example, in the below figure, we select train_processed.csv and the connection manager name is given as “Flat File output”.

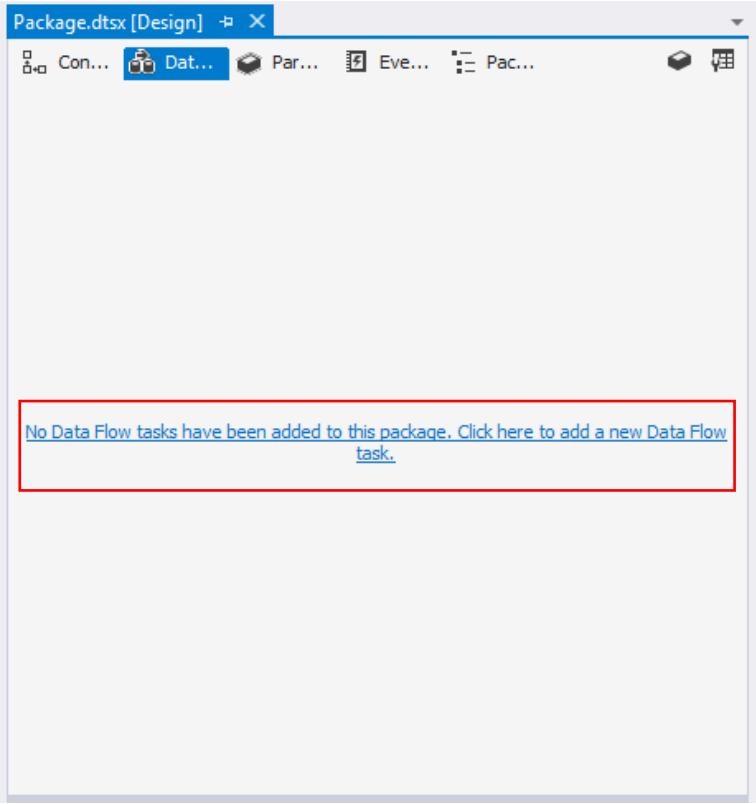


- Increase the length of the Description and the Name column to be longer than 50 characters, and set both to accept Unicode. The screenshot below set the length of Name to 100, proceed to do the same for Description but set the length to 3000. Preview to see if the data has been read in properly.
- Similarly, we put the connection manager name as Flat File train.csv and select the file name train.csv to load.

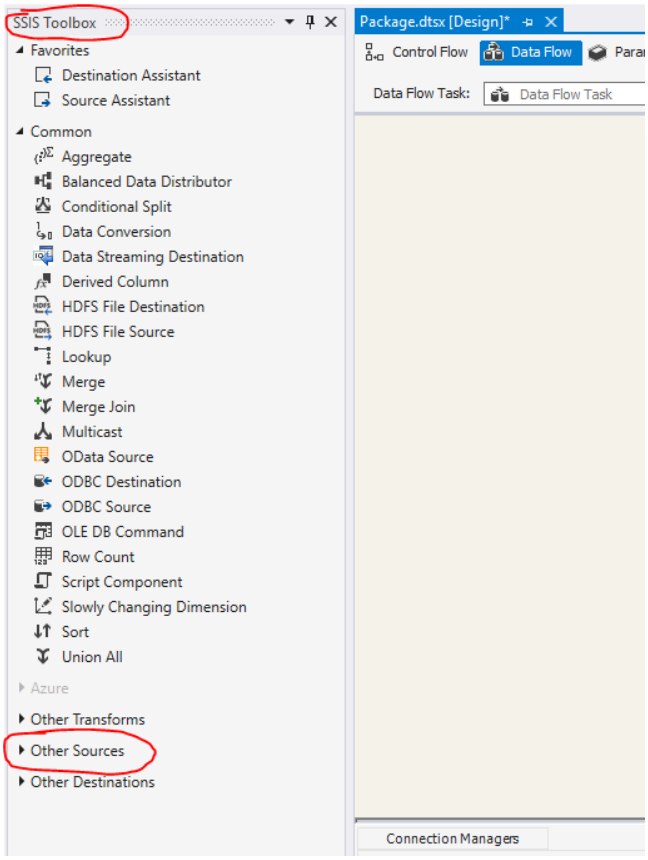


1.6 Create a Data Flow Task

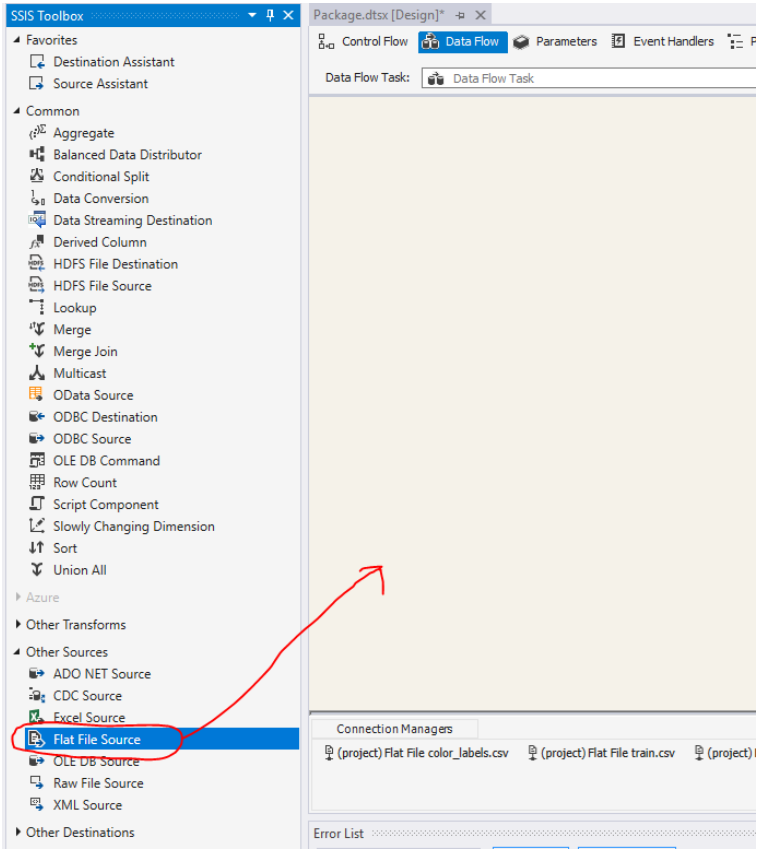
Now select the Data Flow Task from the Package Designer pane.



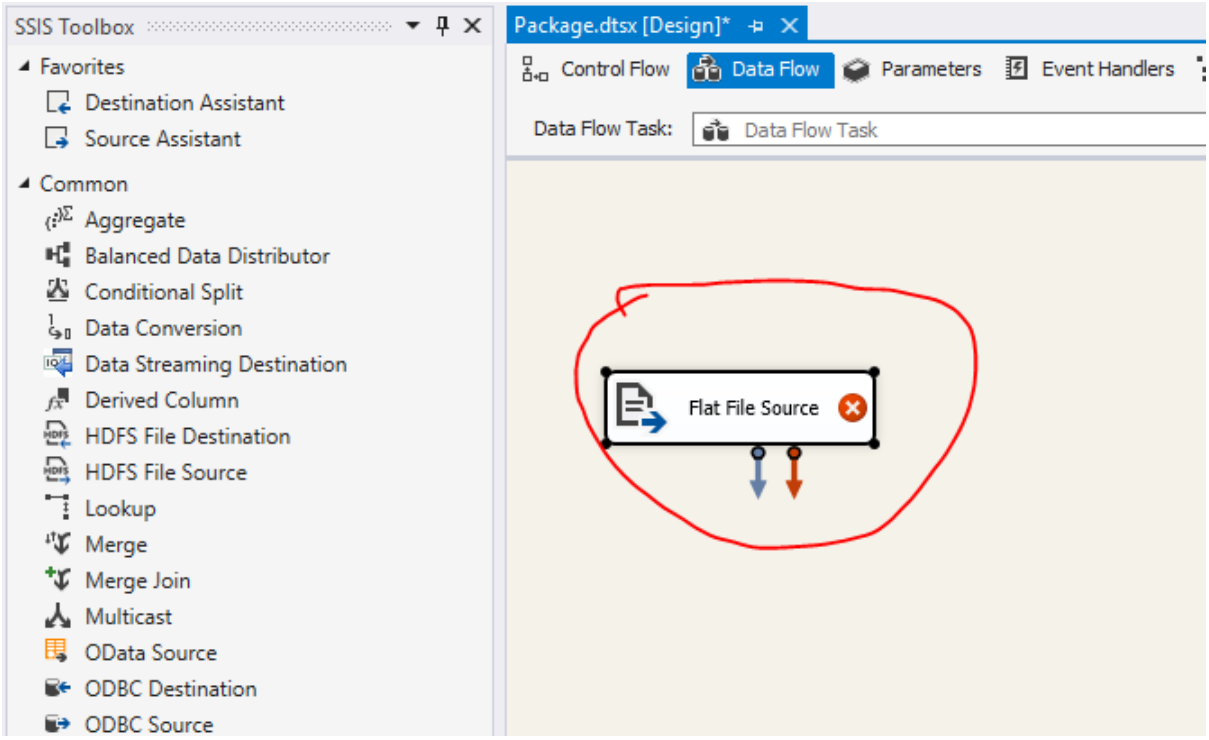
Click on the blue text, navigate the SSIS toolbox on the left, and expand the Other Sources category.



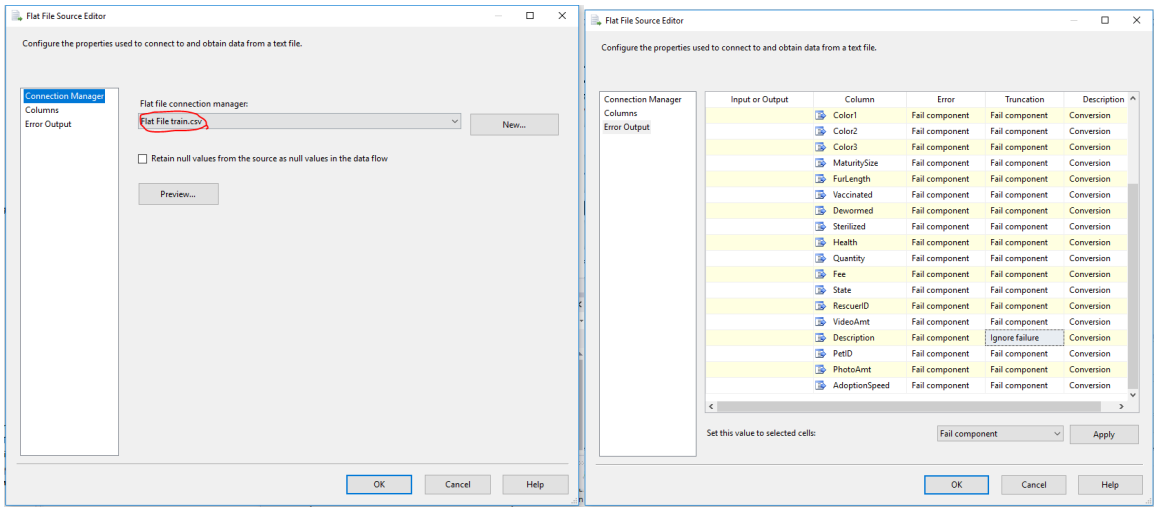
Find Flat File Source, drag and drop it to the empty design pane.



Double click on the Flat File Source icon.



Select the Flat File train.csv Connection Manager from the drop down list (as shown the left screenshot below).

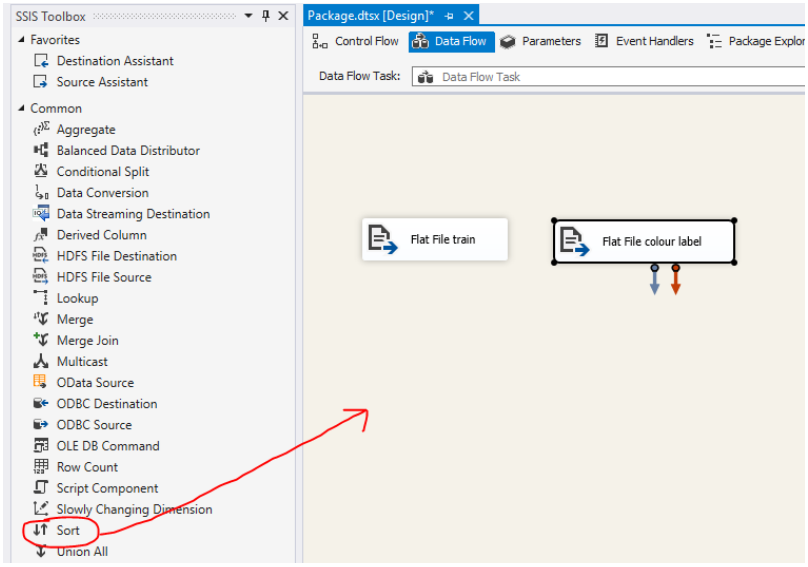


We should also click on the Error Output, to set the truncation error of Description to be ignored as there are a few unnecessarily long descriptions. This is shown in the right screenshot above.

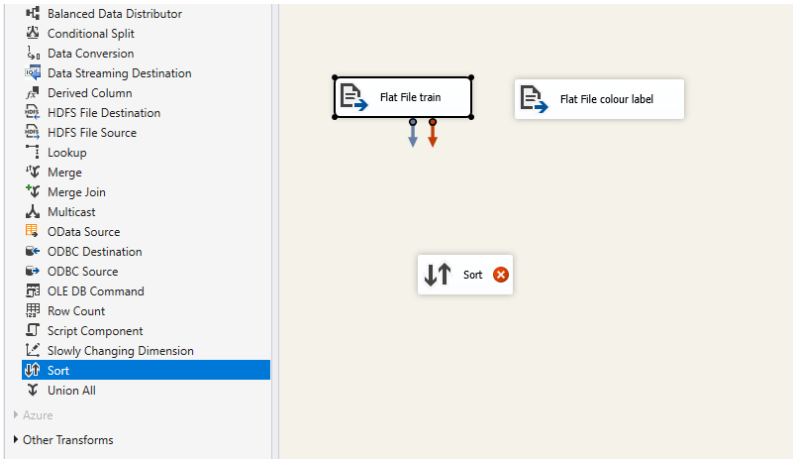
You need to create another Flat File Source for color_labels.csv!

1.7 Sorting

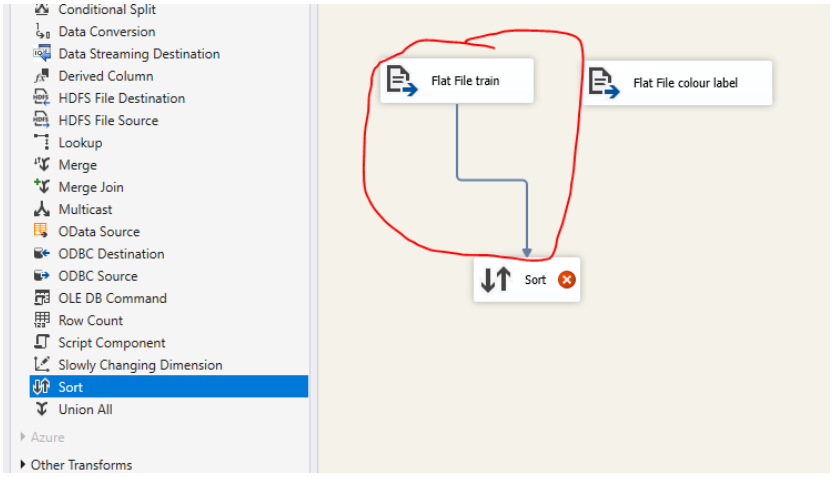
To merge join the two datasets, we need them to be sorted first. Drag and drop the “Sort” tool from under the “Common” category of the SSIS toolbox.



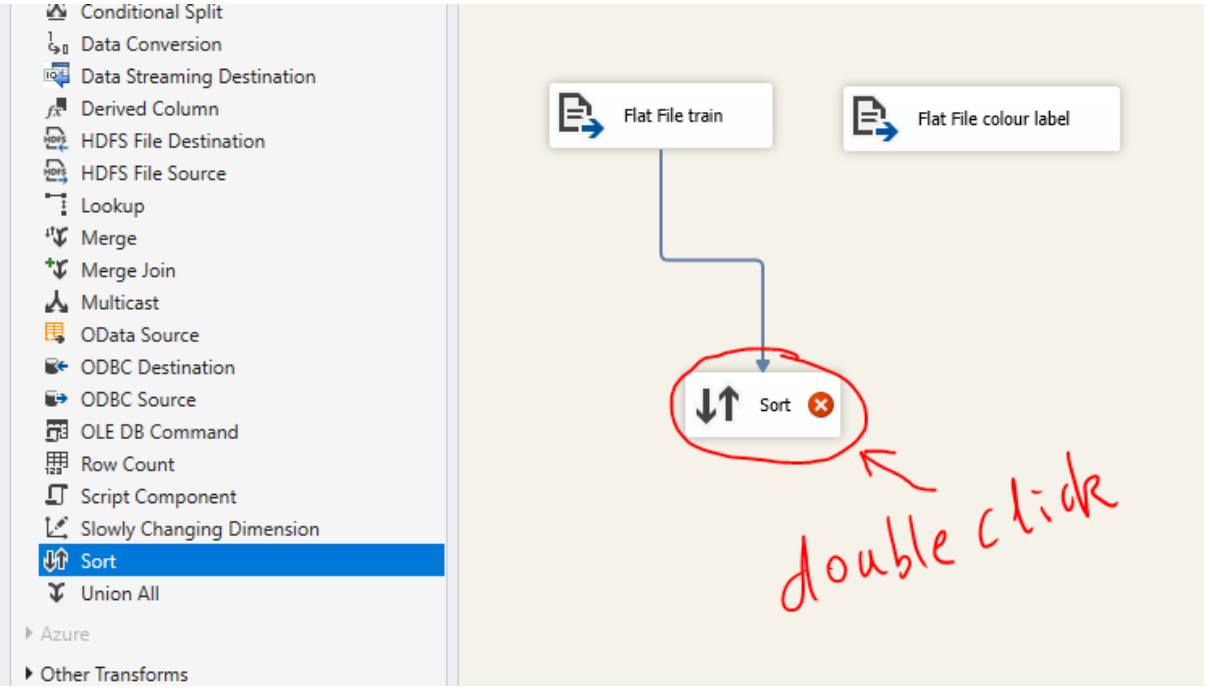
Connect the Flat File corresponding to *train.csv* to Sort.



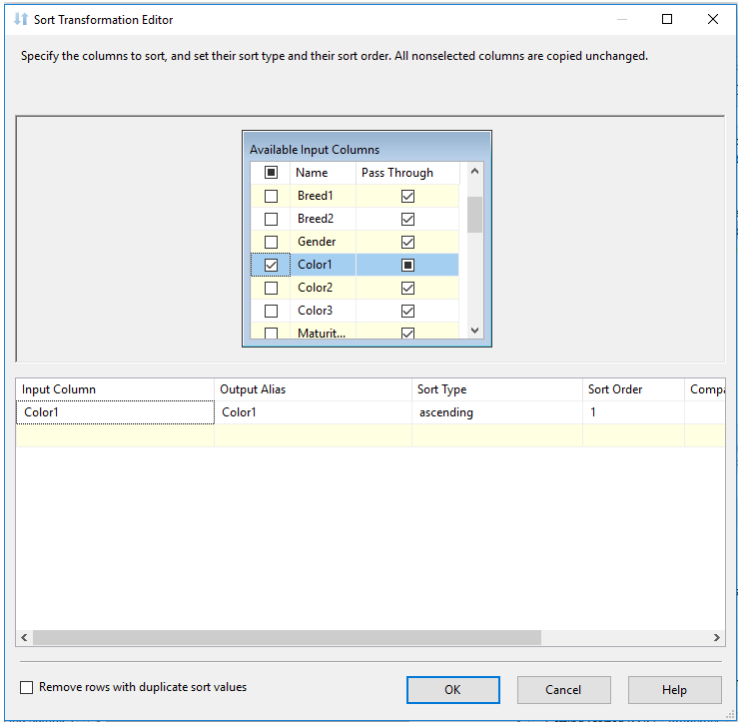
You should see the following screen after the connection.



Double click “Sort”

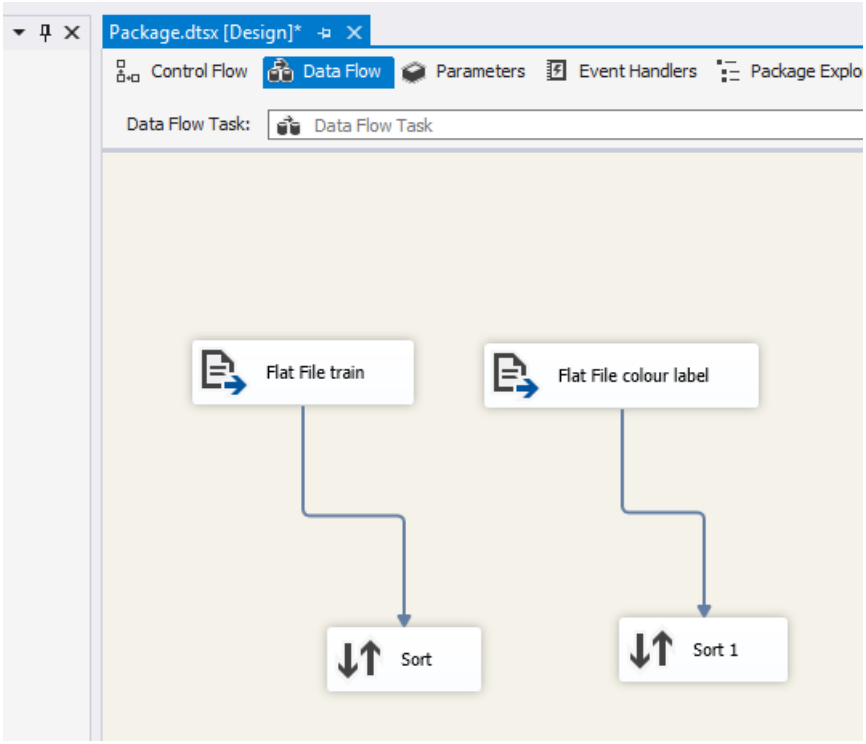


You should bring up the dialog below. Tick **Color1** as the sorting key, as we will use this column to merge.

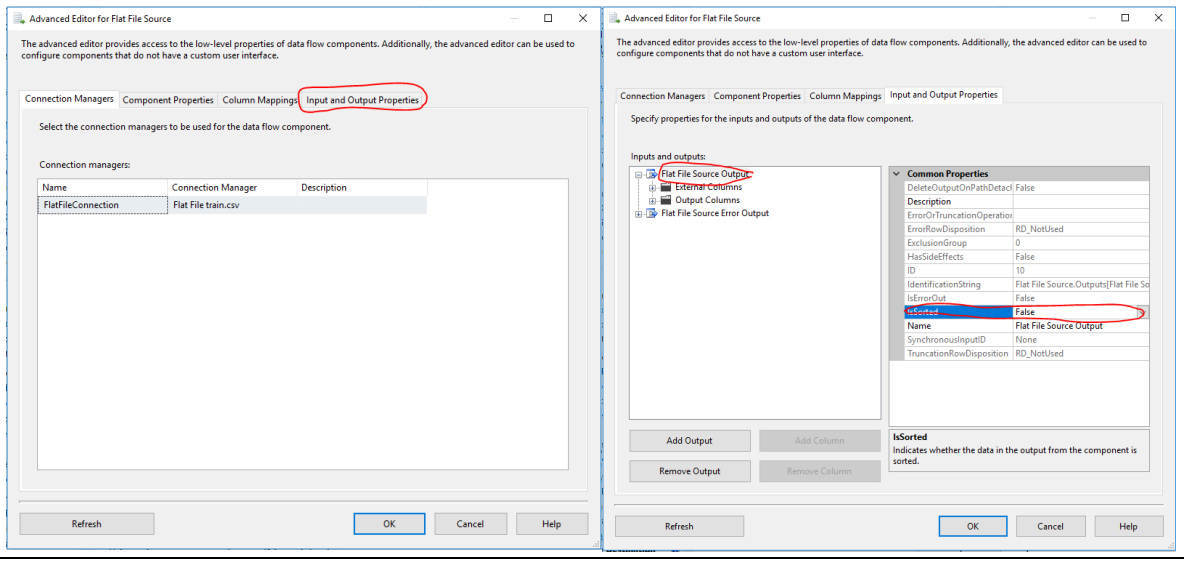


Then do the same for Flat File corresponding to *color_labels.csv* Source.

After the above process, it gives us a processing pipeline like this:

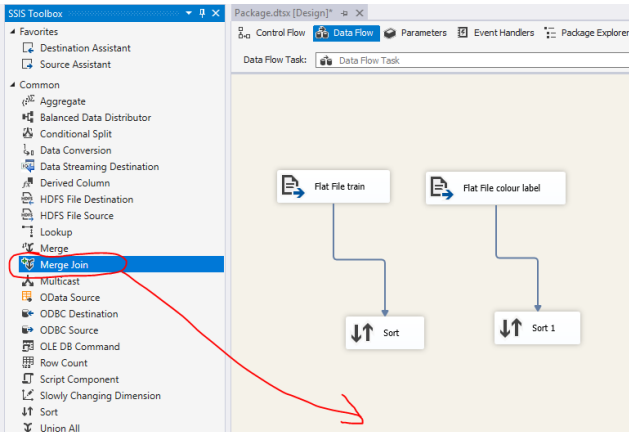


Note: If your data is already sorted, by the color1 column, you need to tell the SSIS so by right clicking on the Flat Data Source icon, and select “Show Advanced Editor”, and then select the Input and Output Properties tab, to set the IsSorted property to true.

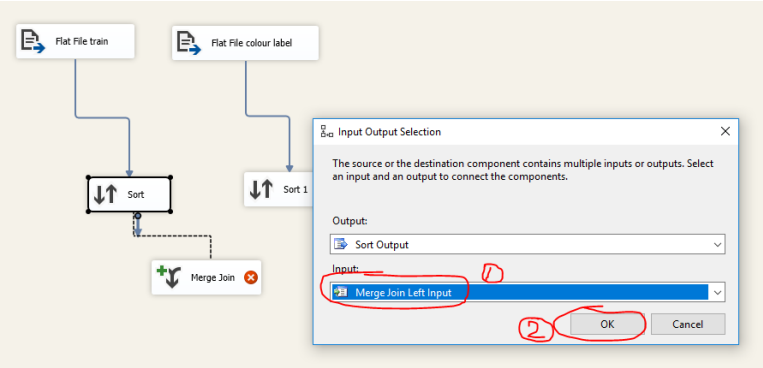


1.8 Merge Join

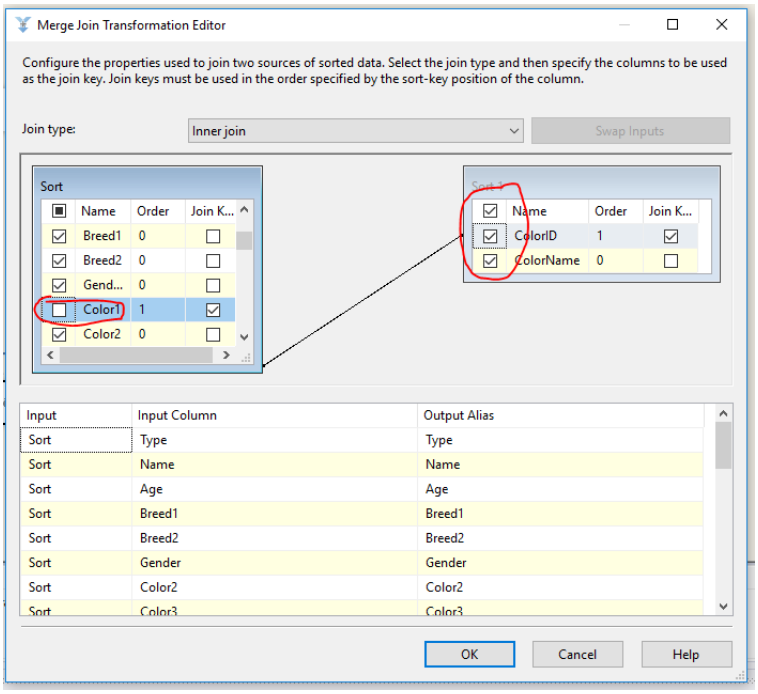
Drag and drop a Merge Join tool.



Get the two sorted outputs connecting with Merge Join.

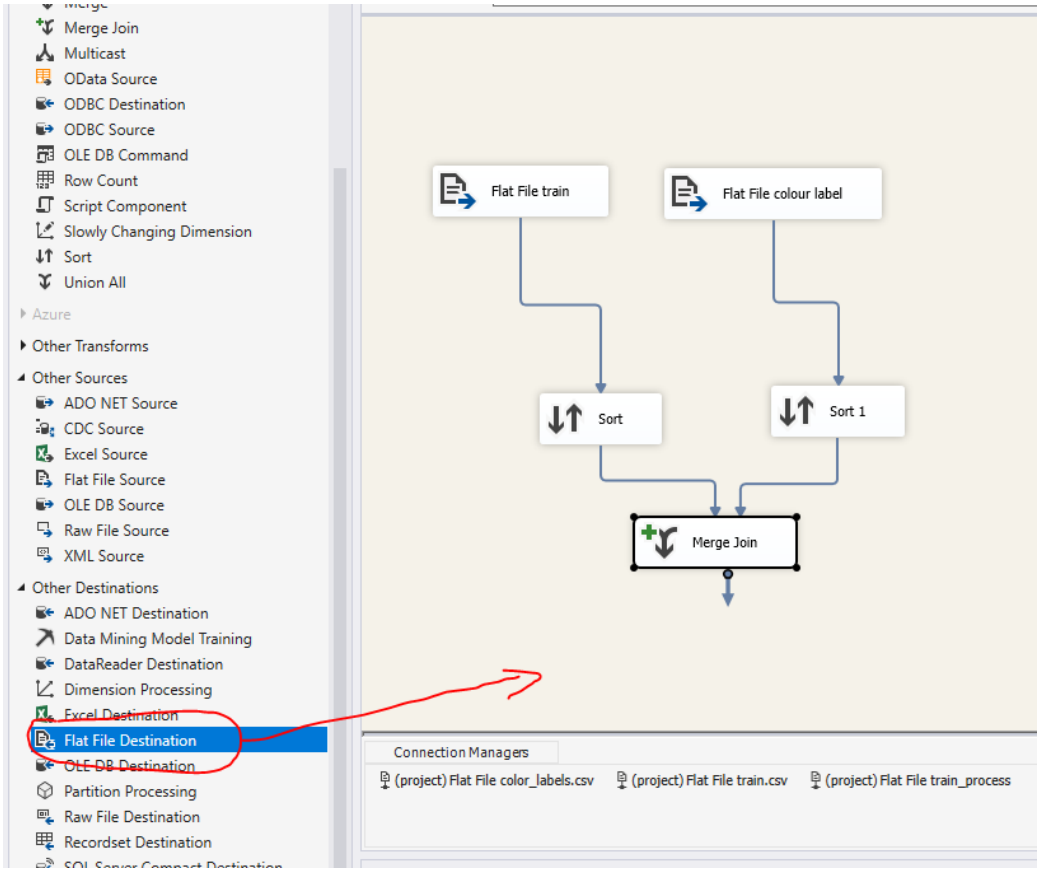


Double click the Merge Join icon allows you to select what to use for the key matching and what columns to keep as the merge output. We select all but the color1 column from the *train.csv* table, and all from the *color_labels.csv* table.

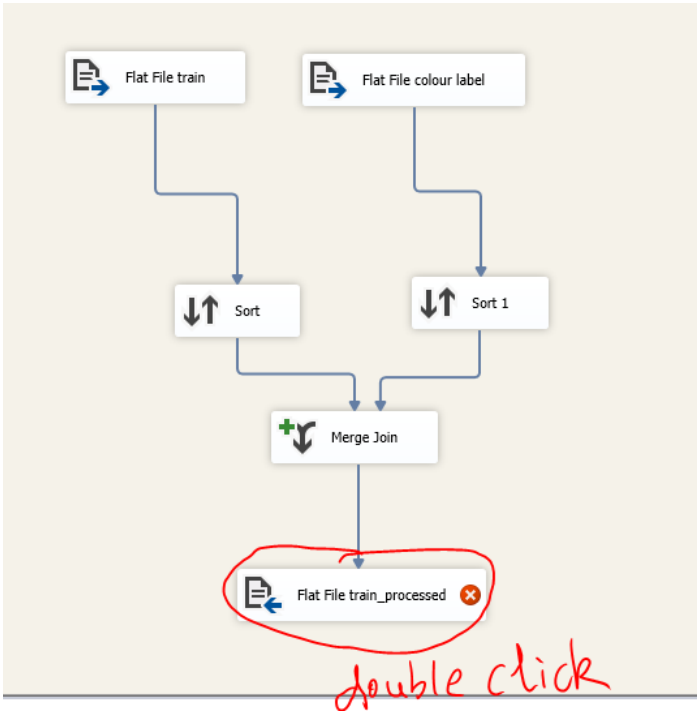


1.9 Output to Flat File Destination

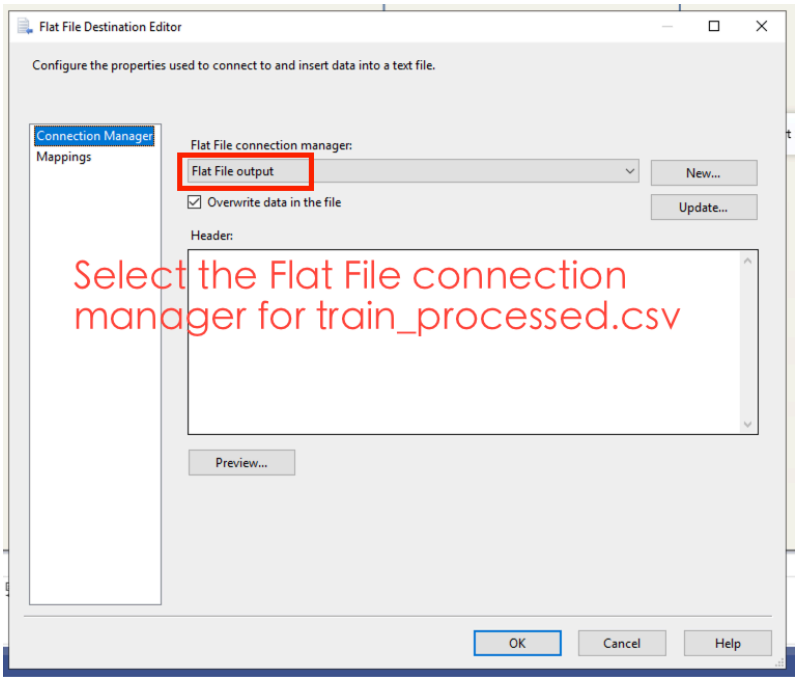
Drag and drop from the SSIS toolbox under the Other Destination category a Flat File Destination icon.



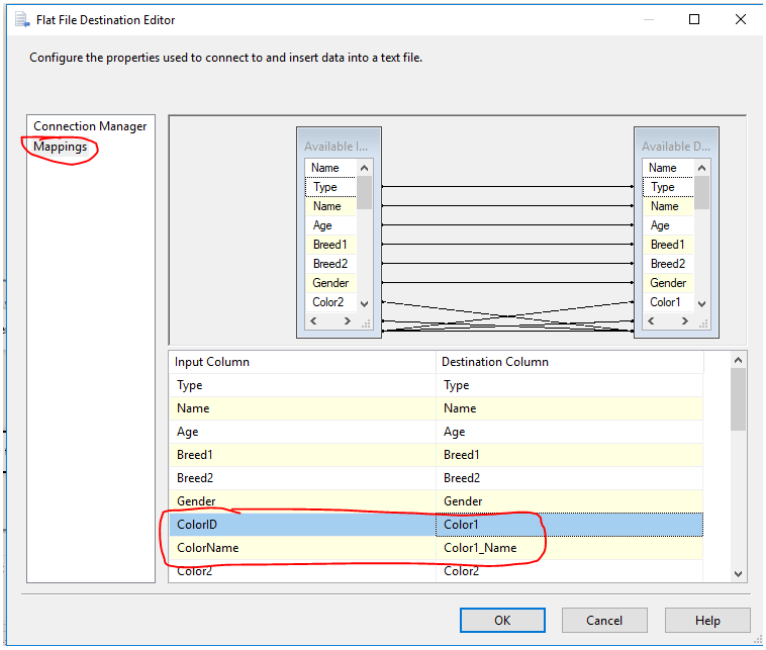
Connect the output of the merge join to the Flat File Destination.



Double click on the Flat File Destination to ensure the data is written the Flat File train_processed.csv connection manager.



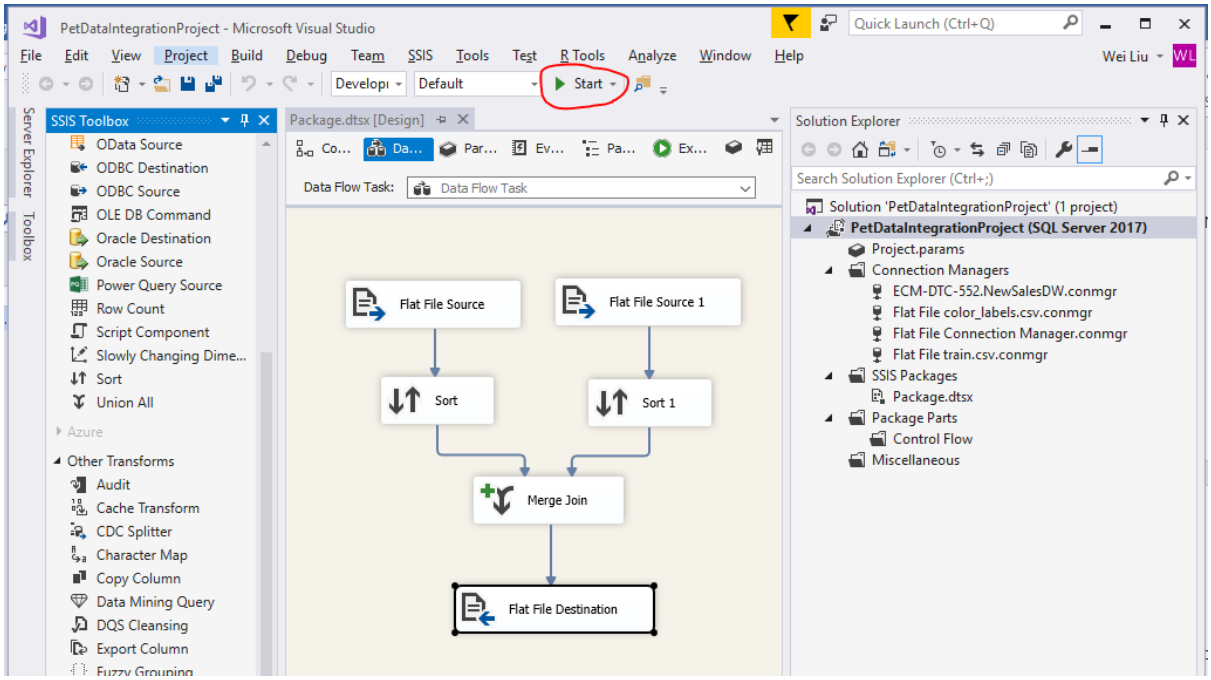
Click “Mappings”, and make sure **Color1** and **Color1_Name** are set to ColorID and ColorName.



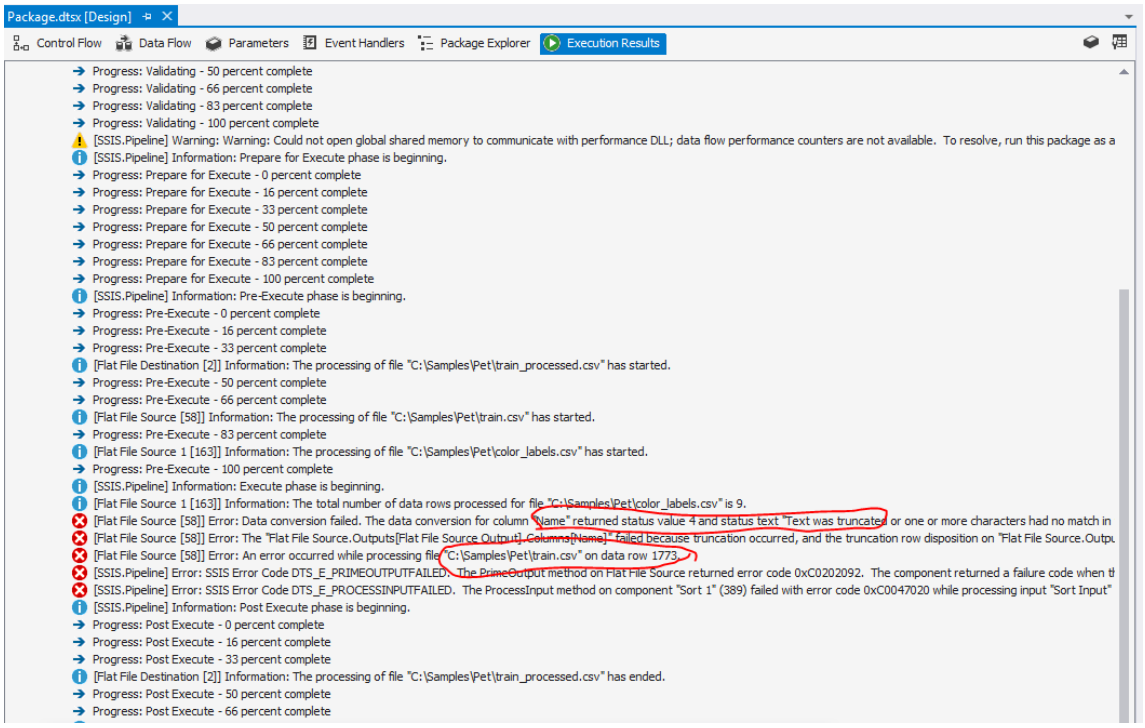
Click “OK”.

1.10 Execute and Trouble Shoot

Now you can click on the Start button to process. If everything goes according to plan, you should see the file train_processed.csv changed with a new column.



Otherwise, you should also learn to troubleshoot by looking at the Execution output:



Look for errors and which row caused error and text truncation in this case, can be solved by increase the length or ignore the error. Note that you may have to redo the downstream pipeline after changing a source input.