**Step 1** – Create metadata schemas of the input files i.e. assign the path of the input files (text files) while creating a metadata of a text file as delimited files. This is done using **tfileInputDelimited** component in Talend.

**Step 2**– Create two local context variables for storing Begin Time and End Time, which will eventually be used for calculating time taken to run this job, in the Contexts Tab in the View Area. (Local context variables are exclusive to this job only)

**Step 3** – Store start date (in milliseconds) in the first local context variable using Java. Develop a logic to store time in milliseconds and embed it within ***tJava*** component. Import necessary packages in the Advanced Settings config bar of ***tJava*** component.

**Step 4** – On successful execution of the component used in Step 3, we need to invoke the previously created metadata schema of the input text files. Hence, transfer the control of data flow to the any one of the Input Files by using an ***OnComponentOK*** trigger on tJava component.

**Step 5** – Unite the two text files and merge them into one single data stream using **tUnite** component.

**Step 6** – Using **tMap** component, embed regex expressions to replace special characters with single spaced strings.

**Step 7** – Invoke **tNormalize** component to break single spaced values to different rows, effectively producing an output stream of single words in each row.

**Step 8** – Invoke the ***tAggregateRow*** component and feed the data flow of previously used ***tNormalize*** component into it. This component will allow configuration changes such as –

1. Applying arithmetic operations on certain columns of the input schema (count() in this case)
2. Grouping the output based on certain columns of the input schema (Column0 or Words, in this case)

**Step 9** – Thrust the resulting data flow into another **tMap** component and filter out null values or blank strings that might have been a result of Steps 5, 6 or 7.

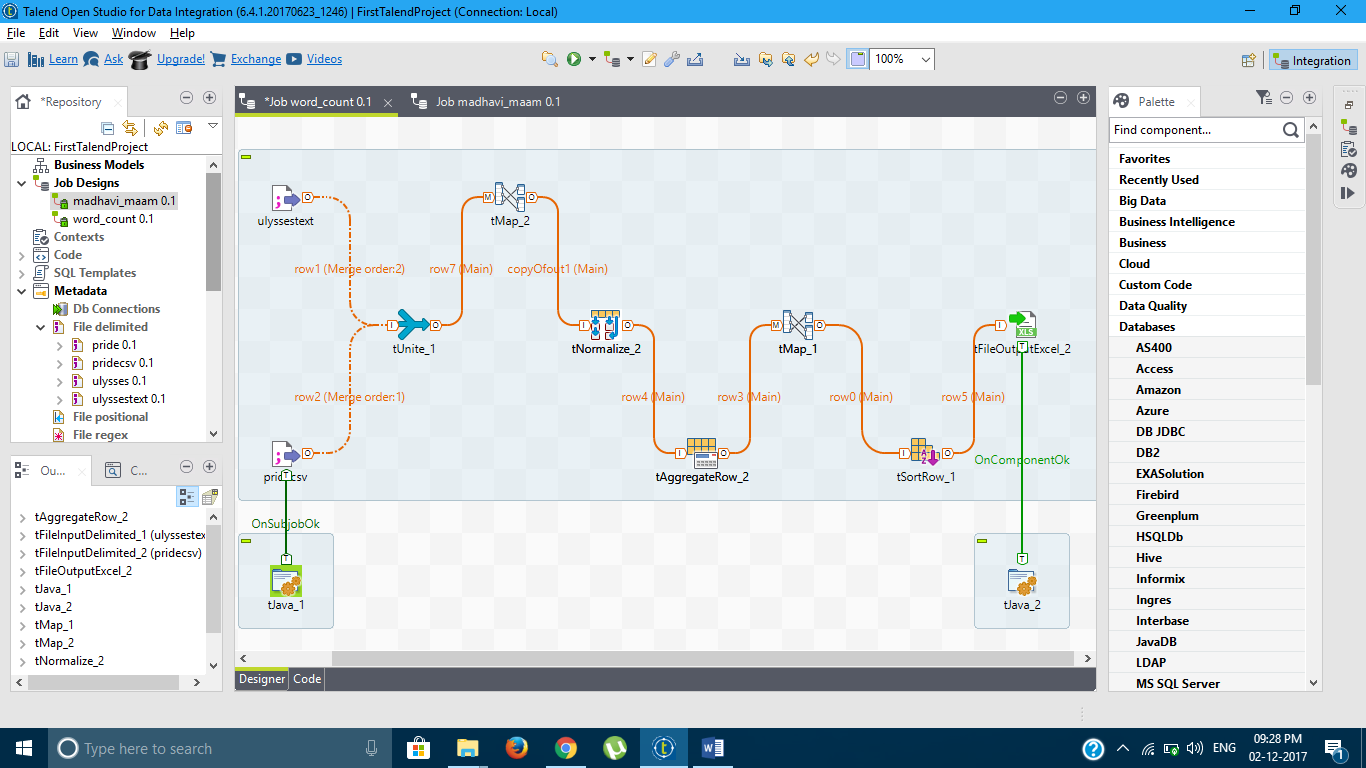
**Step 10** – Pass the data flow to the **tSortRow** component to sort data in descending order based on the word occurrence counts.

**Step 11** – Pass this stream of data now to the **tFileOutputExcel** component to store the final result. This is the final output file. Set the file path accordingly.

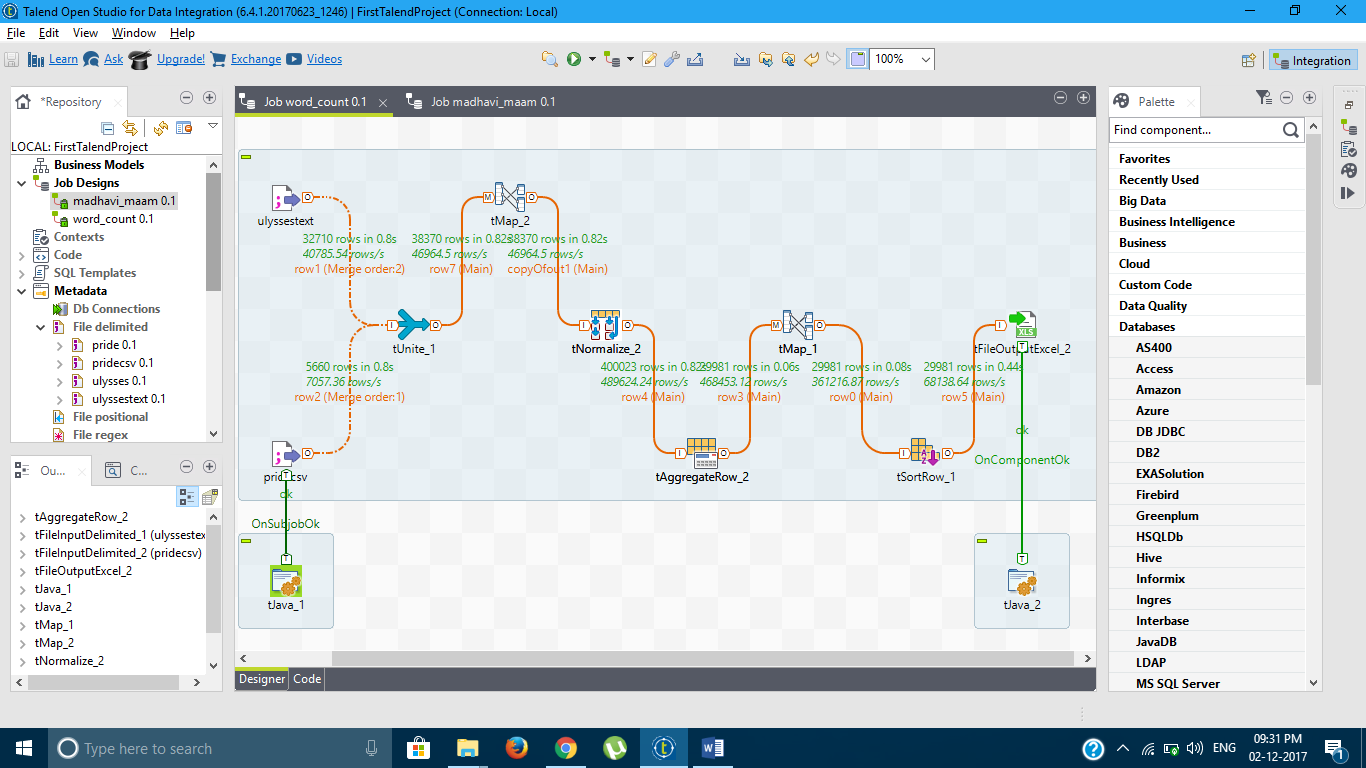
**Step 12** – Build logic in Java, using ***tJava*** Component, to calculate time in milliseconds and store it in the second local context variable. Define the difference between the two context variables as the time taken to execute this Talend Job. Print it on the console.

**Screenshots –**

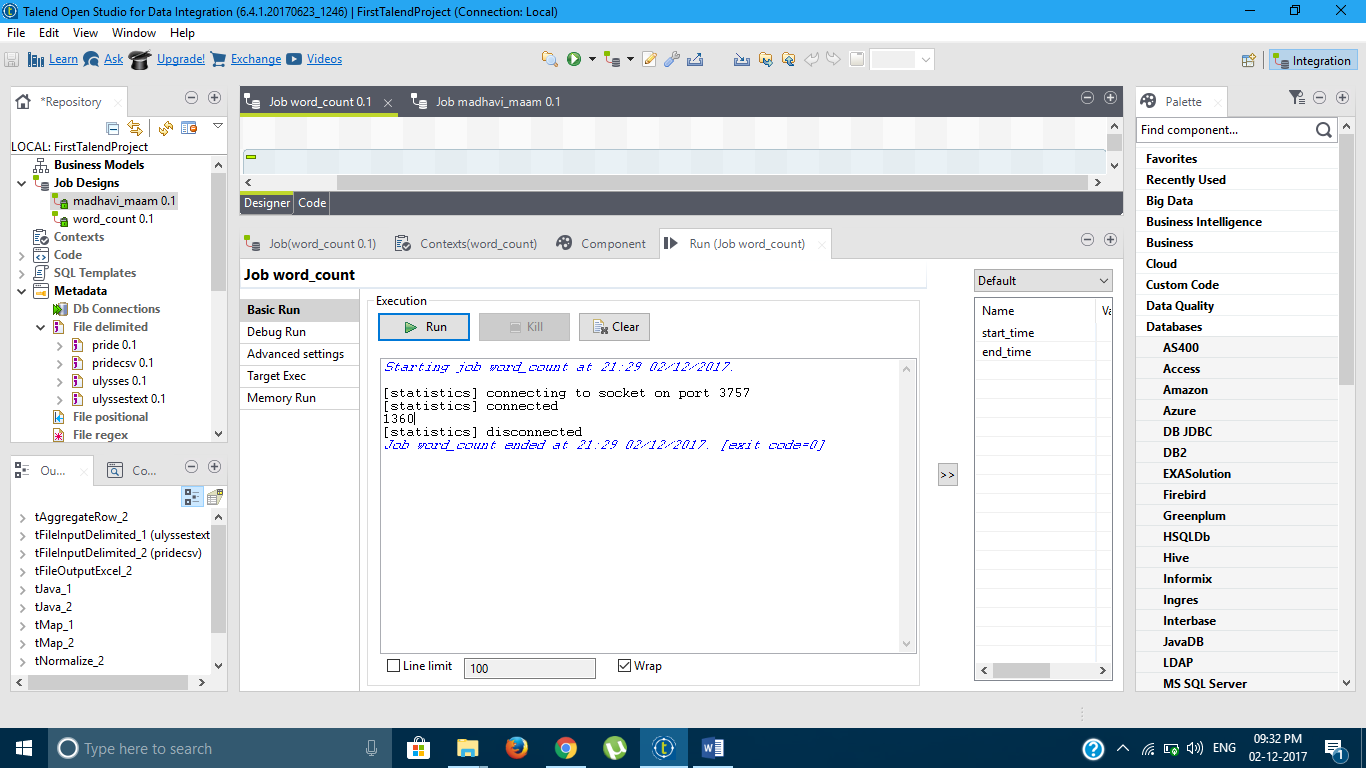
**Pre-Execution:**



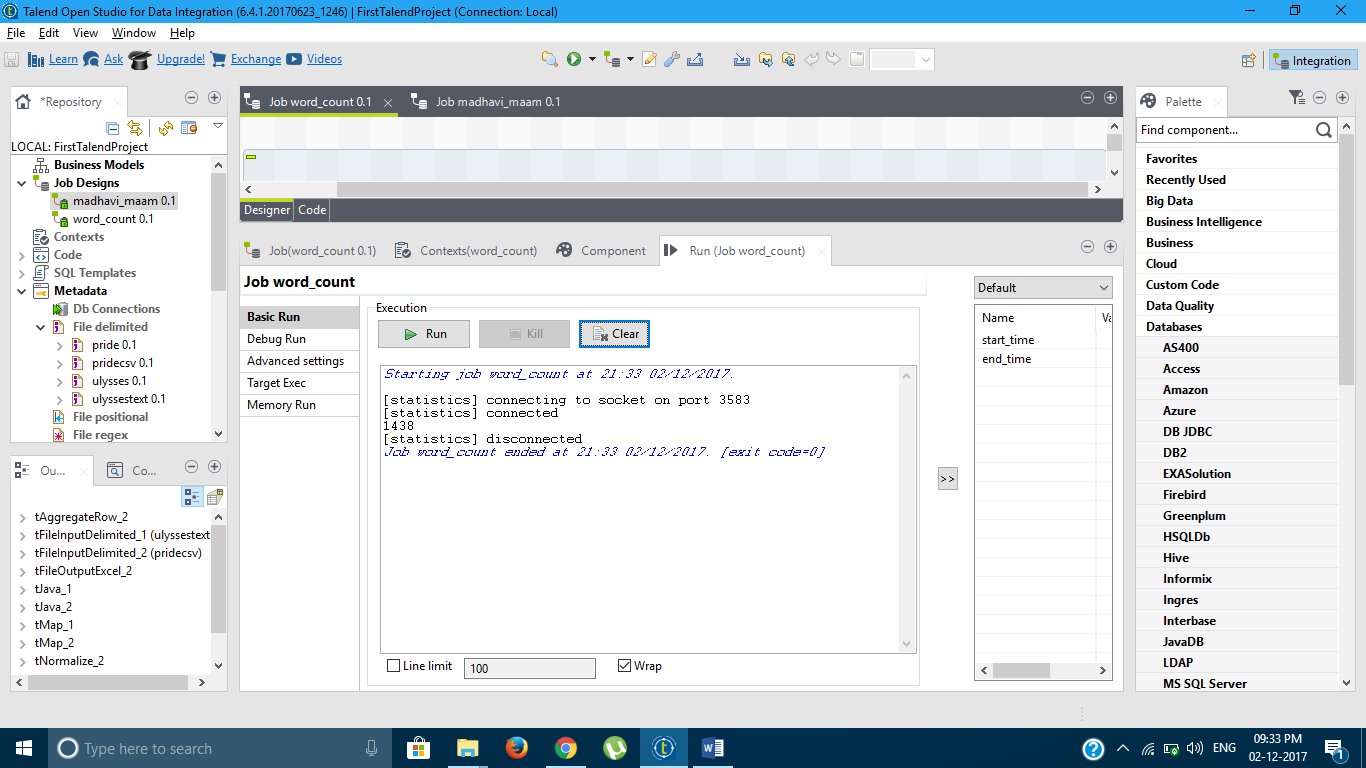
**Post Execution (Data flow mappings and triggers visible)**



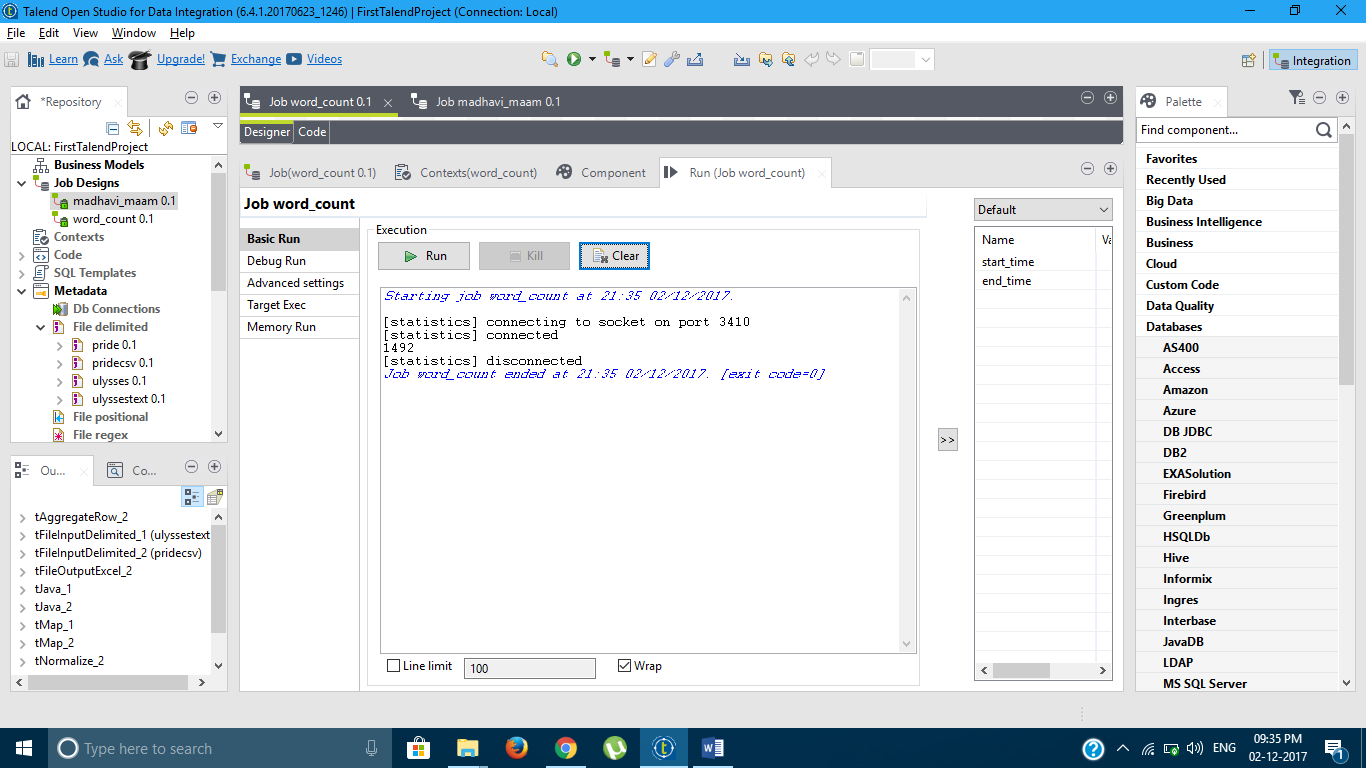
**Console – Execution sample 1 - Time taken 1360 milliseconds**



**Console – Execution sample 2 - Time taken 1430 milliseconds**



**Console – Execution sample 3 - Time taken 1492 milliseconds**



**Console – Execution sample 4 - Time taken 1585 milliseconds**

