Talend Open Studio Quick Start Guide

Talend provides an easy to use graphical interface for creating workflows for BigData ETL (Extract, Transform and Load). Using the HDP components Talend enables users to import raw data into Hadoop, create and manage schemas using HDP’s HCatalog and includes Pig and Hive platforms for analyzing these data sets.

Talend Open Studio enables an enterprise to work with existing data and existing systems, and use Hadoop to power large scale data analysis across the enterprise.

The document will guide you through the installation of Talend Open Studio, an add-on for HDP and will guide you writing your first Talend job to get data into Hadoop cluster. The Talend job will then be modified to do data analysis using Pig.

# Starting Talend Open Studio for HDP

Step 1: Download and launch the application

* Download the Talend Open Studio add-on for HDP from <TODO>.
* After the download is complete, unzip the contents in an install location.
* Invoke the executable file corresponding to your operating system.

Step 2: In the License window that appears read and accept the end user license agreement to continue. The startup window appears as shown:



* Create a new project by providing some project name (for example HDPIntro) and clicking the “Create…” button. Click “Finish” on the “New Project” dialog.
* Select the newly created project and click “Open”.
* The “Connect To TalendForge” dialog appears, you can choose to register or click “Skip” to continue.
* A progress information bar and a welcome window displays consecutively. Wait for the application to initialize and then click “Start now!” to continue.

Talend Open Studio (TOS) main window appears and is now ready for use.



# Pre-requisites

* Make sure the HDP cluster is up and running.
* Make sure the user launching the TOS has appropriate permissions on the HDP cluster. Example if hdptestuser is the one launching TOS, then run the following commands on the gateway machine as the administrator user (hdfs) to create home directory if one doesn’t exist already for hdptestuser:

% hadoop dfs –mkdir /user/hdptestuser

% hadoop dfs –chown hdptestuser:hdptestuser /user/hdptestuser

# Creating the first job

Using the TOS we will design a simple job of transferring a file into the Hadoop cluster.

Step 1: Create a new job

In the Repository tree view, right-click the Job Designs node and select Create job from the contextual menu.



In the “New Job” wizard provide a name (for example HDPJob) and click “Finish”. An empty design workspace opens up corresponding to the Job name.

Step 2: Build the job

Jobs are composed of components that are available in the Palette.

* Expand the “Big Data” tab in the Palette and click on the component “tHDFSPut” and click on the design workspace to drop it there.
* Double-click tHDFSPut to define component in its “Basic Settings” view”. Set the values in the Basic Settings corresponding to your HDP cluster.



* Since the above component depends on the file /tmp/input.txt, create the file with following contents:

101;Adam;Wiley;Sales

102;Brian;Chester;Service

103;Julian;Cross;Sales

104;Dylan;Moore;Marketing

105;Chris;Murphy;Service

106;Brian;Collingwood;Service

107;Michael;Muster;Marketing

108;Miley;Rhodes;Sales

109;Chris;Coughlan;Sales

110;Aaron;King;Marketing

Step 3: Run the job

Voila! You have a working job. You can run it by clicking the green play icon. You should see the following:



To verify the operation, run the following command on your HDP cluster.

[hdptestuser@hdp ~]$ hadoop dfs -ls /user/hdptestuser/data.txt

Found 1 items

-rw-r--r-- 3 hdptestuser hdptestuser 252 2012-06-12 12:52 /user/hdptestuser/data.txt

So the local file gets created successfully on the HDP.

# Modify the Job to do Data Analysis

Now that the data is in, we will use Pig to aggregate the data.

* Expand the Pig tab in the Big Data Palette and click on the component tPigLoad and place it in the design workspace.
* Fill the values for the different fields as shown, make sure the namenode URI and the JobTracker host corresponds to your HDP cluster. The Input File URI corresponds to the path where we previously imported the file.



* Define the schema of the data being loaded into Pig. Click on the “…” button corresponding to “Edit schema”. In the Schema dialog enter the schema of the input data as shown:



* Connect the components to define the flow. To connect the two components, right-click the source component (tHDFSPut) on your design workspace, select Trigger > On Subjob Ok from the contextual menu, and click the target component.



* Add the component tPigAggregate next to tPigLoad. Connect the components, right-click on tPigLoad, select Row > Pig Combine from the contextual menu, and click on tPigAggregate.



* Double-click tPigAggregate to define the component in its Basic Settings view. Click on the “Edit schema” button and define the output schema as shown:



* Add a column to “Group by”, choose “dept”. In the Operations table, choose the “people\_count” in the Additional Output column, function as “count” and input column “id” as shown:



* Add the component tPigStoreResult next to tPigAggregate. Connect the components, right-click on tPigLoad, select Row > Pig Combine from the contextual menu, and click on tPigStoreResult.



* Double-click tPigStoreResult to define the component in its Basic Settings view. Specify the result directory on HDFS as shown:



* Job is ready to be run. Save the job and click the play icon to run.

After the job runs you can see the output at:

[hdptestuser@hdp ~]$ hadoop dfs -cat /user/hdptestuser/output/part-r-00000

Sales;4

Service;3

Marketing;3