



Academy of Engineering

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Talend

Documentation

Title: Context and Global Variables

Submitted By

Shivangi Pandey	BDSA24
Rakshanda Mulay	BDSA25
Harshali Dube	BDSA30
Shweta Madge	BDSA40

**DEPARTMENT OF COMPUTER
ENGINEERING MIT ACADEMY OF
ENGINEERING ALANDI (D), PUNE**

1. Context Variables

Variables represent values which change throughout the execution of a program. A global variable is a system variable which can be accessed by any module or function. It retains its value after the function or program using it has completed execution.

A context variable is a variable which is defined by the user for a particular context.

Depending on the circumstances the Job is being used in, you might want to manage it differently for various execution types, known as contexts. For instance, there might be various testing stages you want to perform and validate before a Job is ready to go live for production use.

The purpose of the context variables is to have variables which are transferred between tasks within the same task hierarchy. A context variable will always belong to one context. When the execution thread terminates, the context variables are automatically deleted.

This means that one task can add a context variable, and another task (within the same context) can read and/or modify the context variable.

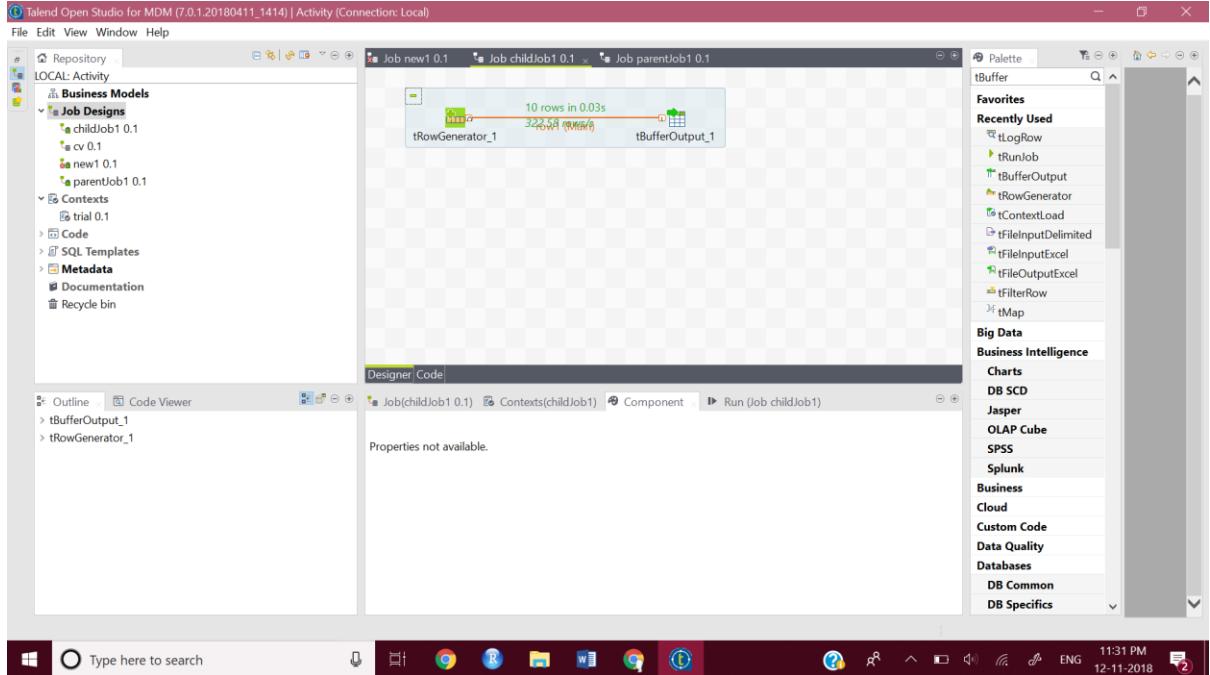
2. Returning value from child job to parent job

Steps:

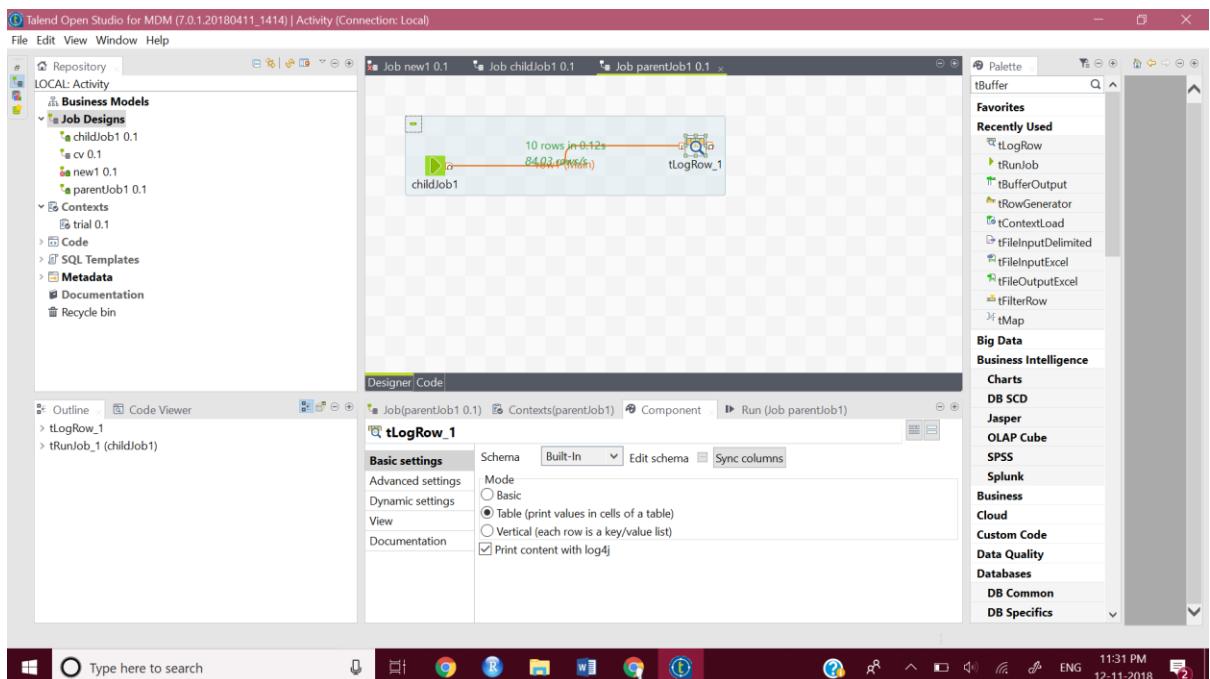
1. To create the first subjob i.e child job, add a **tRowGenerator** in the design workspace, which has a talend sample data and a **tBufferOutput** to it. Connect both with **tLogRow** component.

2. Run child job.

Context Variables

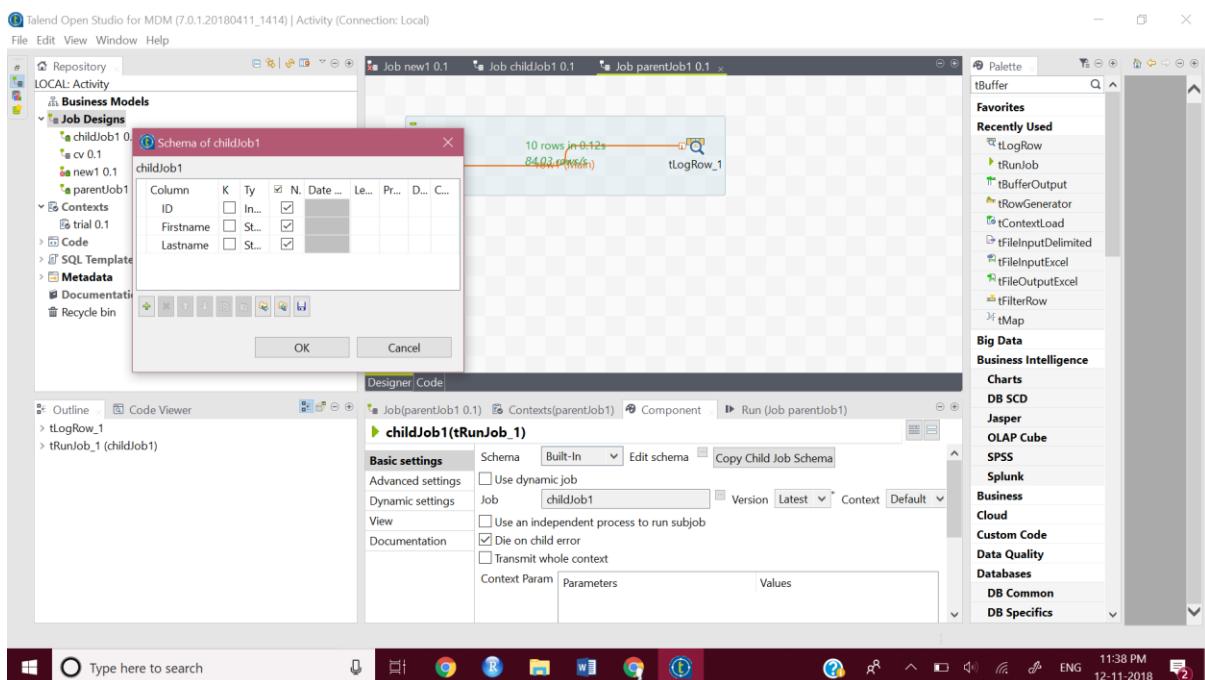
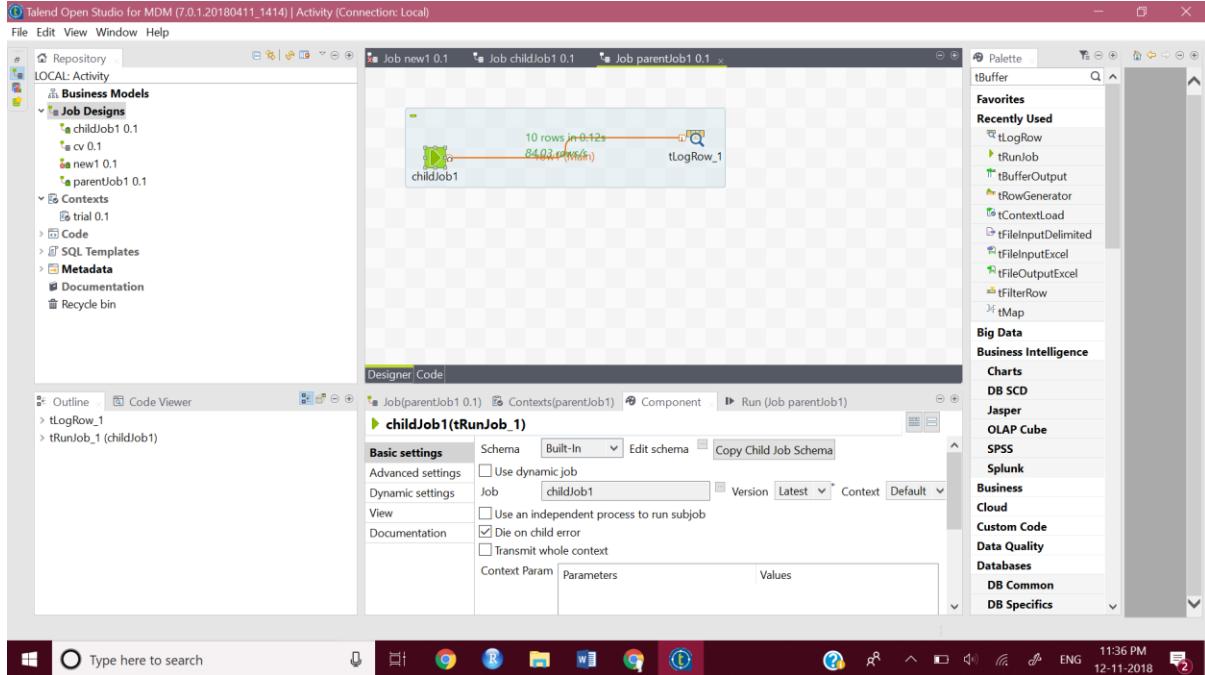


3. To create the second job i.e. parent job, drag the child job created in the design workspace and also tLogRow to print the output to console. Connect using row main.



4. To make the child job data available to parent job, right click on child job and click on settings. Use “copy child job schema” option.

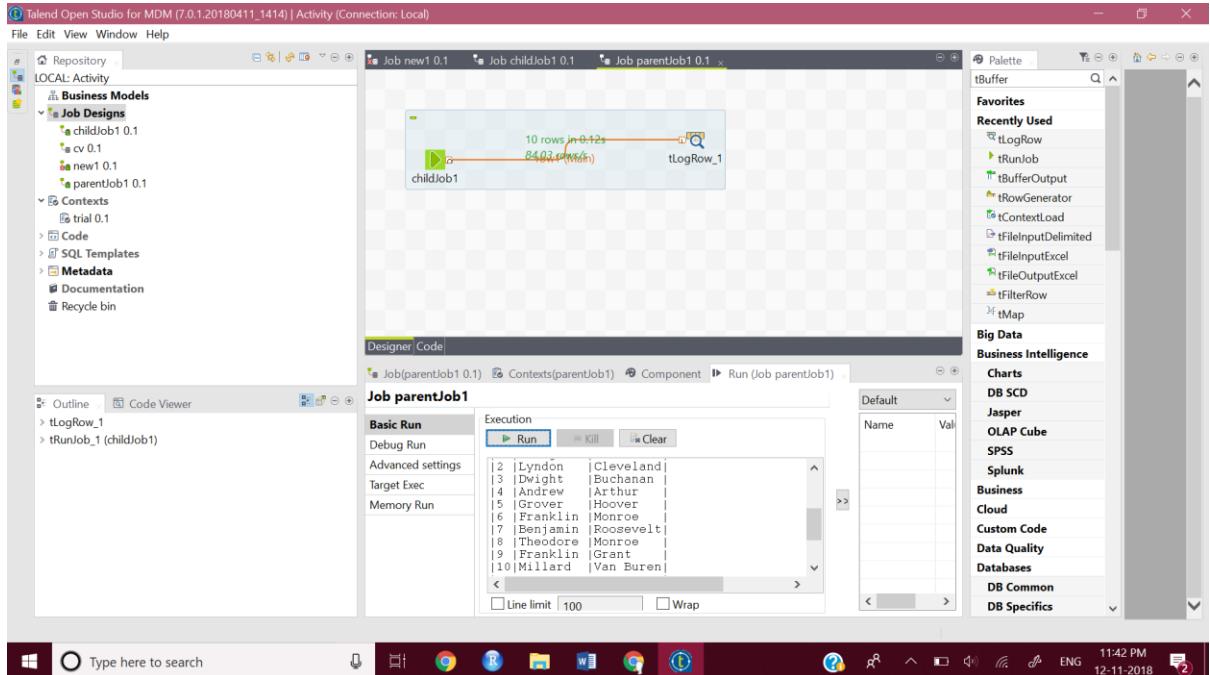
Context Variables



4. Run Parent job.

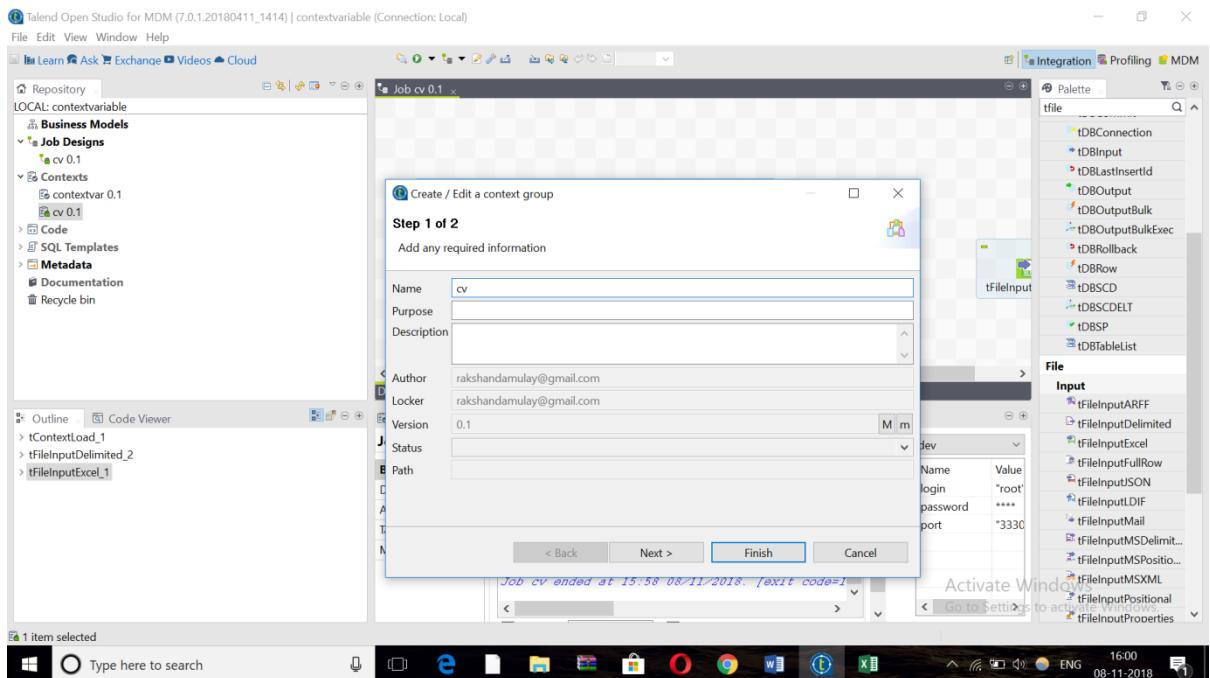
5. You will get the output as shown in the screenshot below.

Context Variables

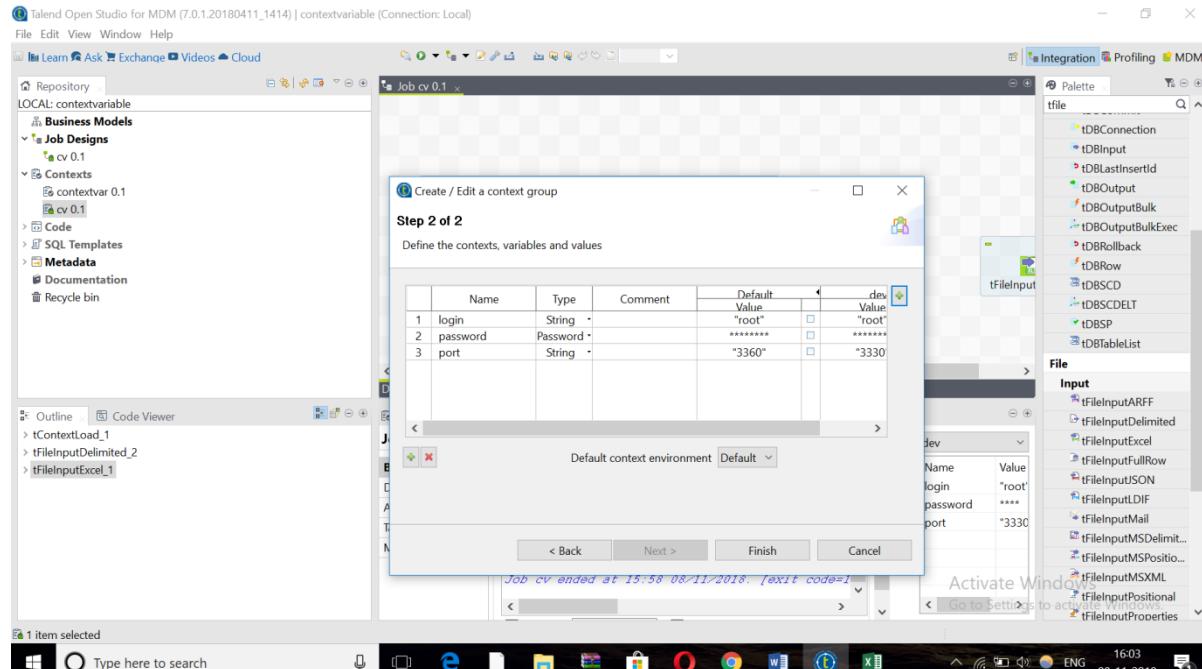


Creating a context group

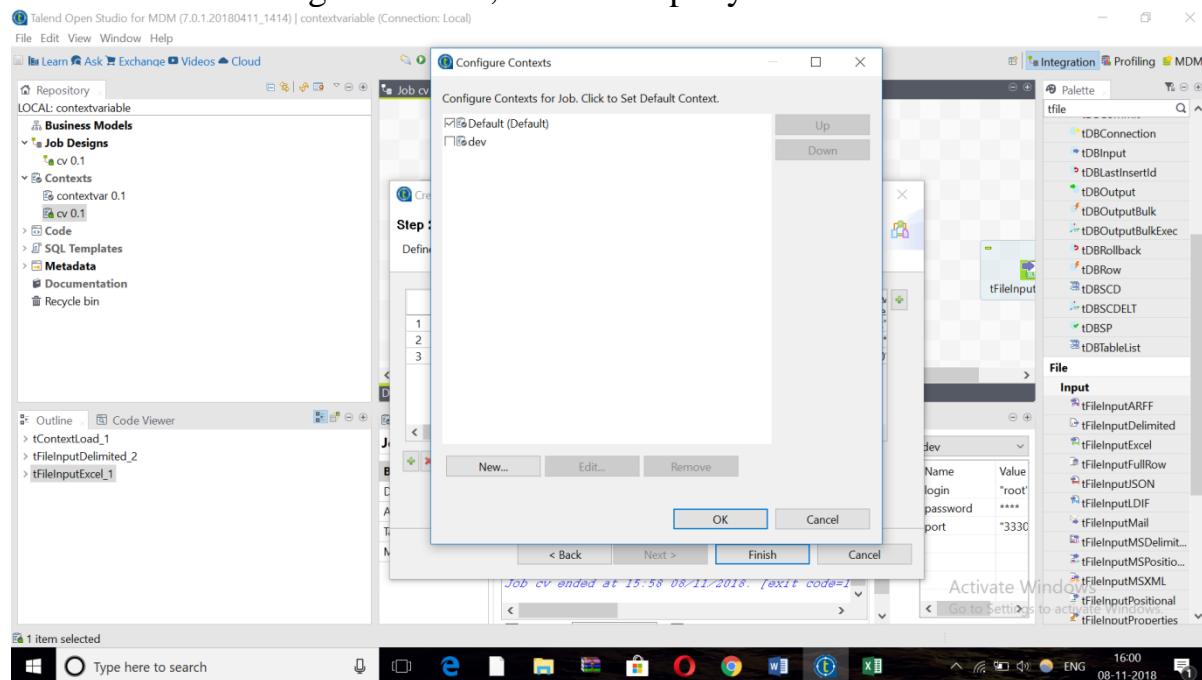
1) Firstly, Open the repository in which select the option of context and right click on it. You will find the various options, in which select create context. As shown below give any name to your context and click on next.



- 2) Then click on plus sign and add variables with its value. Check whether the context environment is default or not.

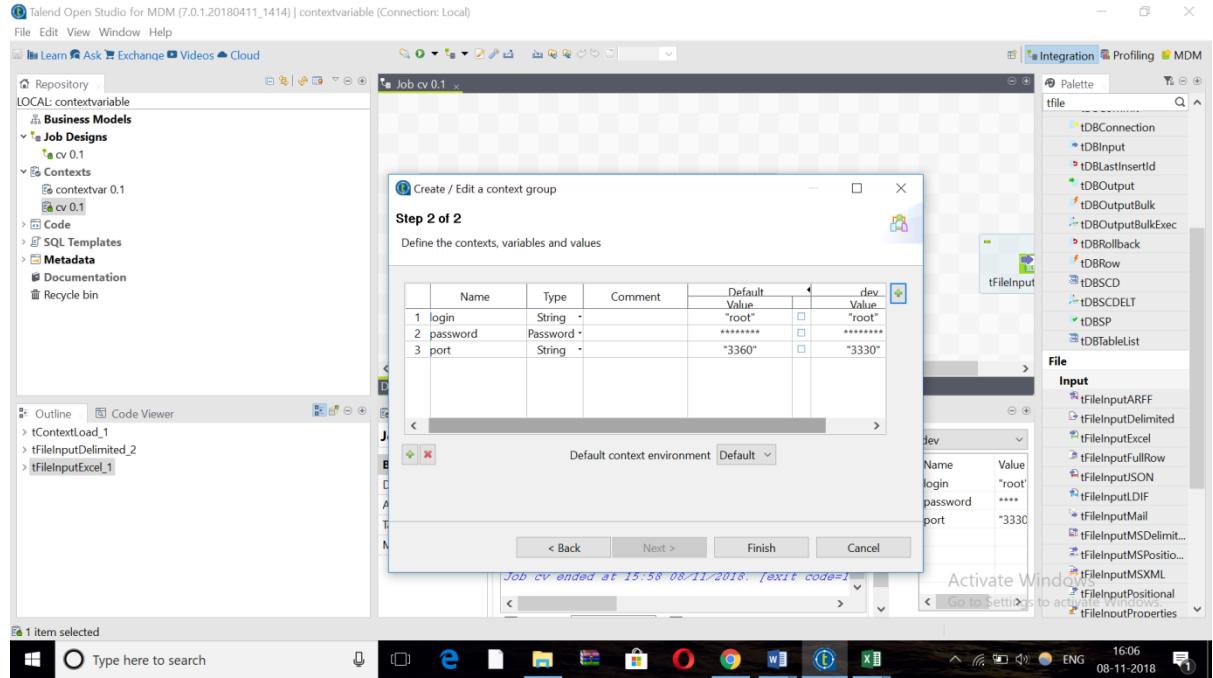


- 3) You can add different context by clicking plus sign present on the top right side and then make changes like edit, remove as per your needs.



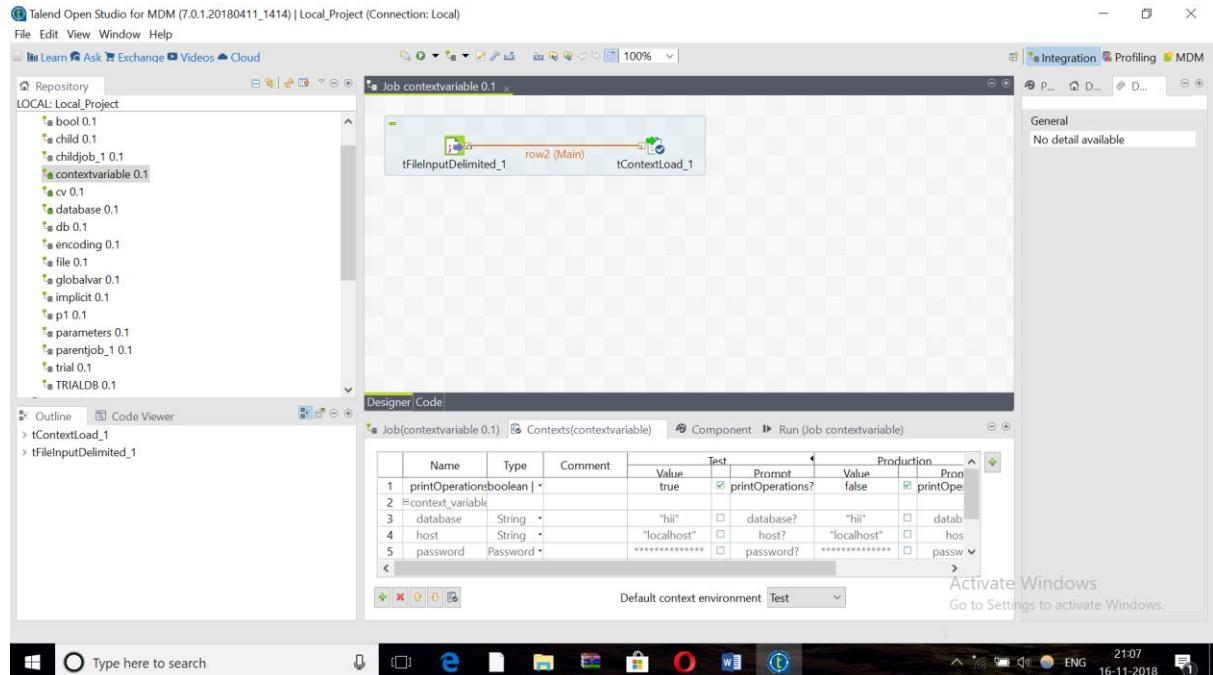
- 4) You can give different values to your variables for specific context

Context Variables



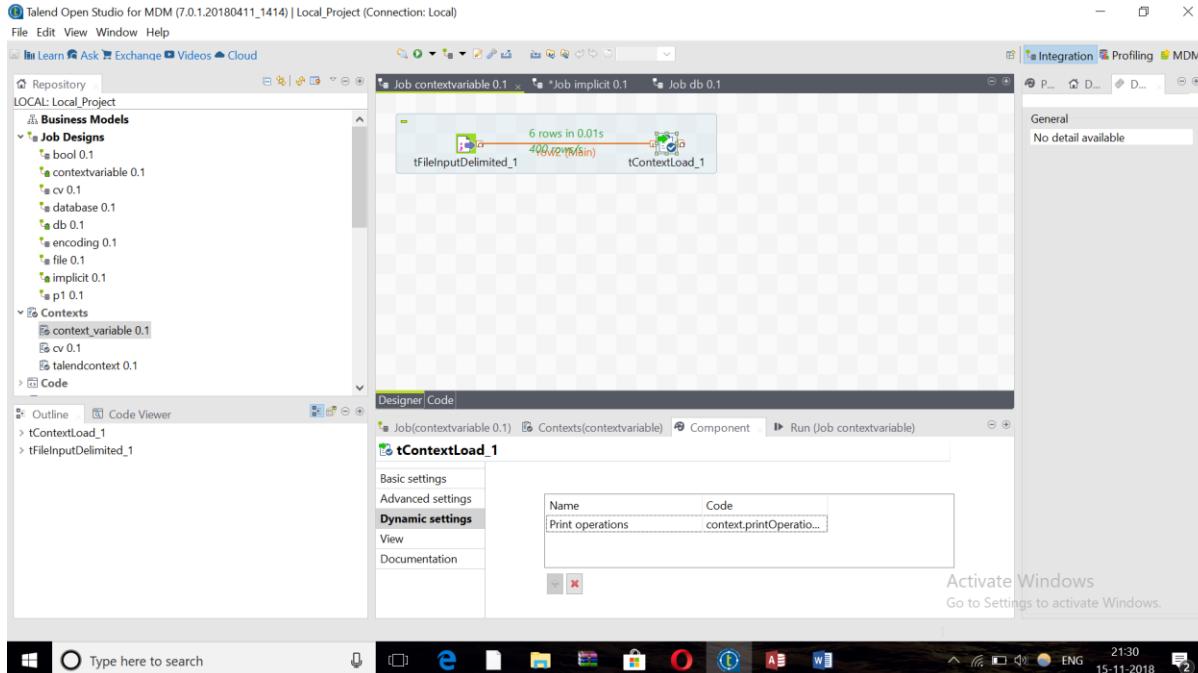
This about loading the file dynamically:

1. Firstly, add tfileinputdelimited and tcontextload. Connect both of them with row main

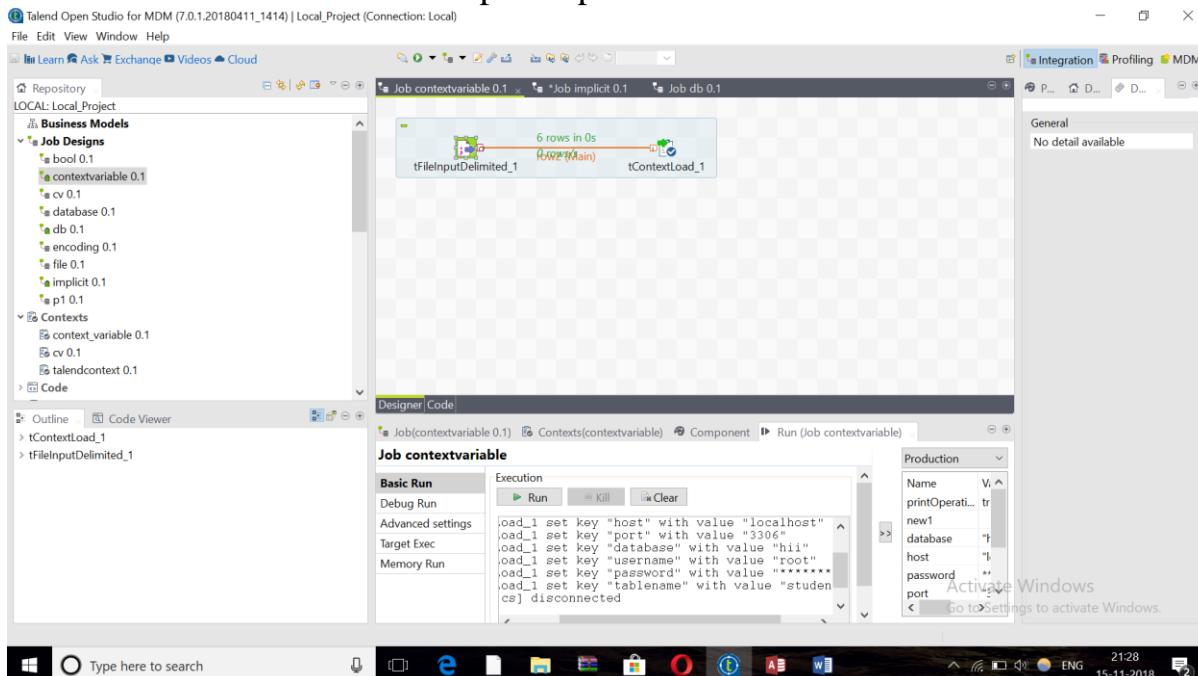


2. Select the any text file separated with ; After selecting the file add the context variables of two different environments. Make printOperations true for one environment and false for another environment.

Context Variables

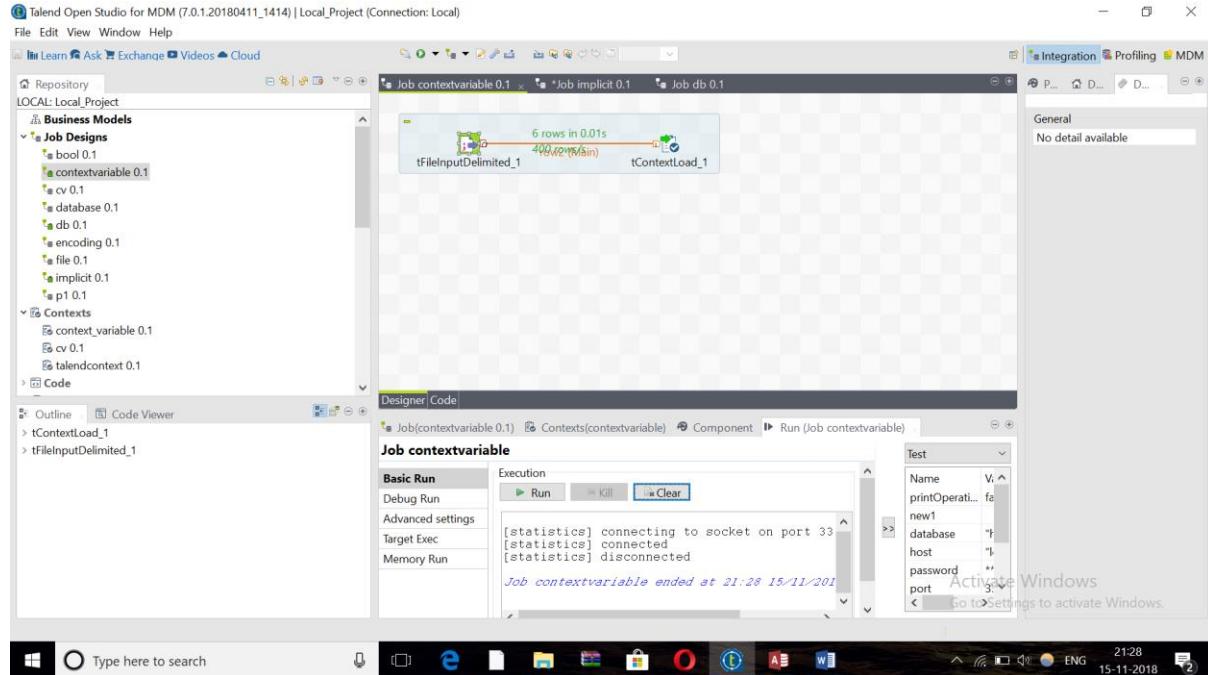


3. Run the job and you will get the values of the files displayed in the environment which was selected true for the print operations.



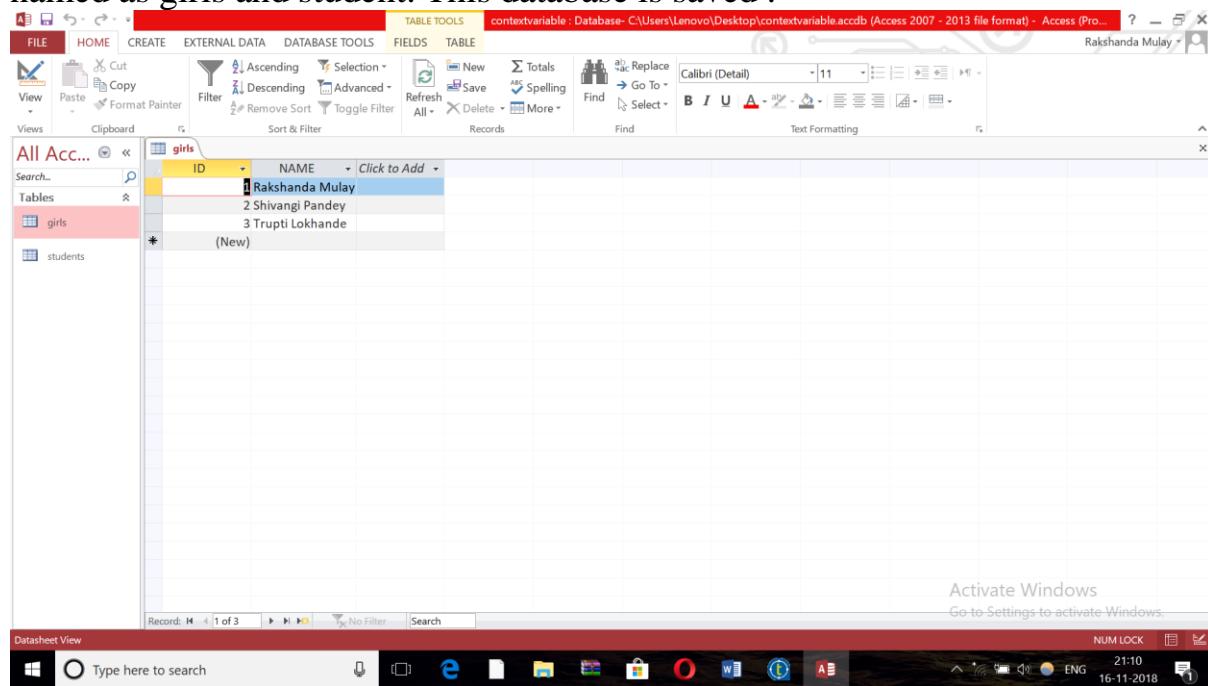
4. Again when you select other environment you will not get any printed operations.

Context Variables



To use the context variable through database.

1. To use context variables through database, you must have any database server installed. Since Microsoft access is more flexible, we have created two tables in it named as girls and student. This database is saved .



Context Variables

Activate Windows
Go to Settings to activate Windows.

2. After saving database, you need to create new context group with the variable shown the picture.

While assigning the values make sure that you are passing the correct values in quotes. Here also we have created two environments named as test and production.

Step of 2

Define the contexts, variables and values

	Name	Type	Comment	production	test
1	username	String		"rakshandamulay@gmail.com"	"rakshandamulay@gmail.com"
2	password	Password		*****	*****
3	tablename	String		"students"	"girls"

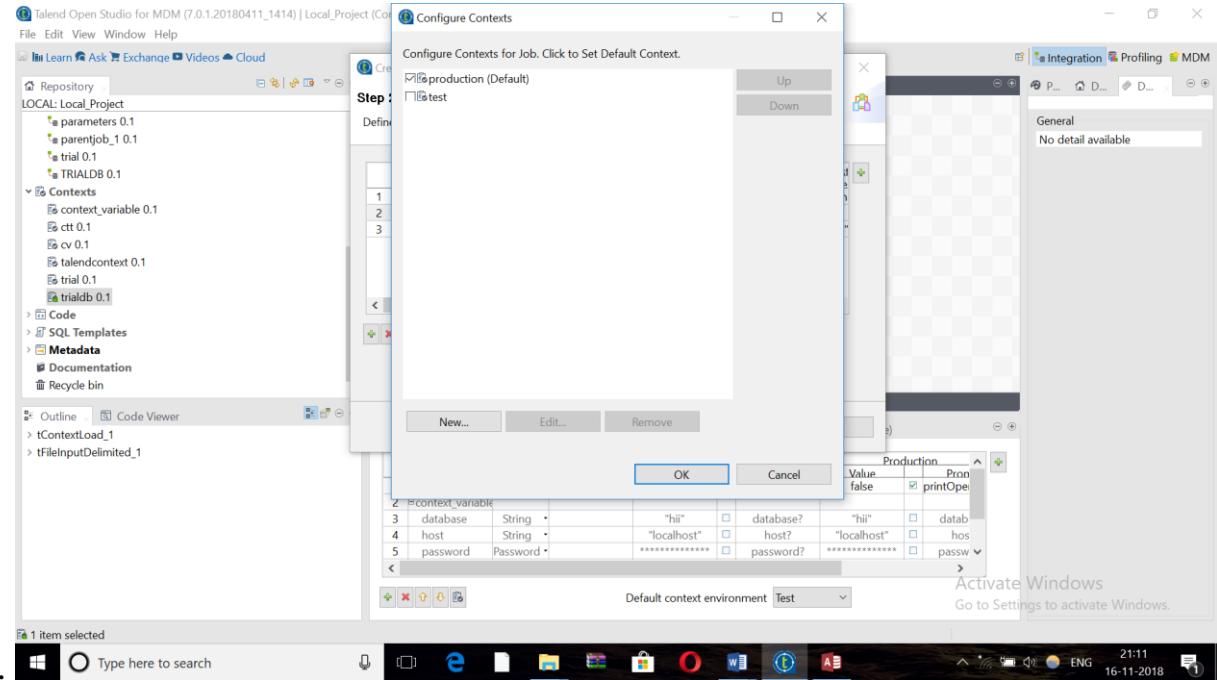
Default context environment production

	Name	Type	Comment	Test	Promot	Value	Production	Promot
1	printOperationsboolean	Boolean		Value	checked	printOperations?	Value	checked
2	context_variable	String		Value	checked	context_variable	Value	checked
3	database	String		Value	checked	database?	Value	checked
4	host	String		Value	checked	host?	Value	checked
5	password	Password		Value	checked	password?	Value	checked

Default context environment Test

Activate Windows
Go to Settings to activate Windows.

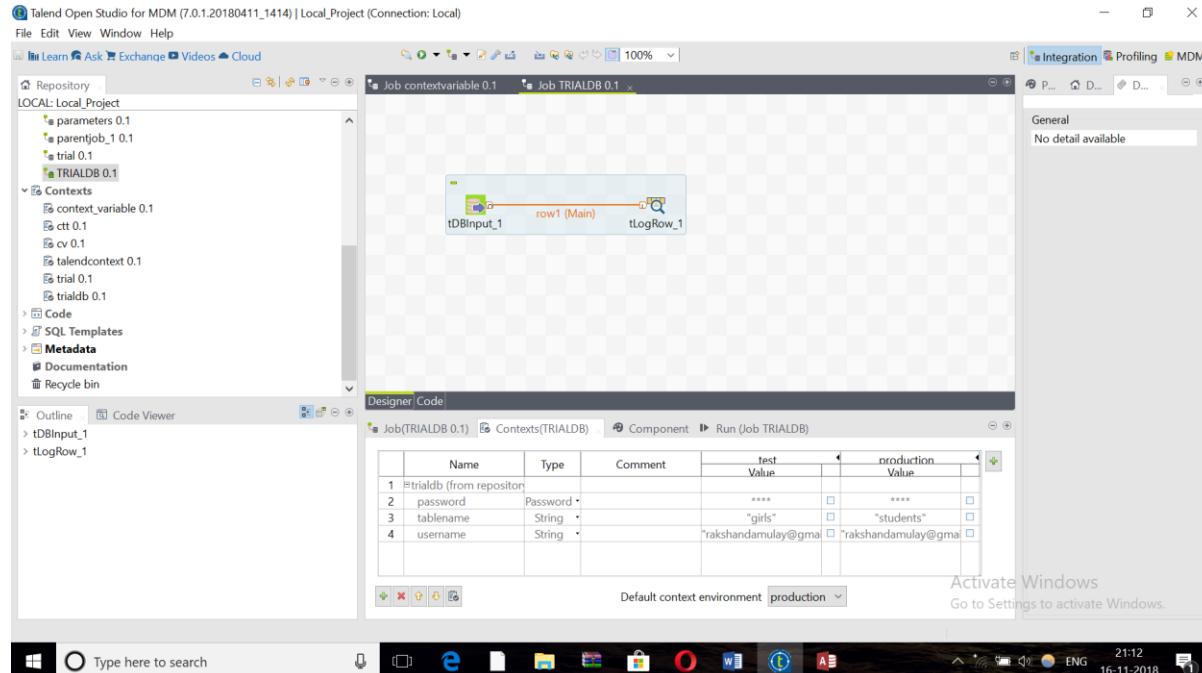
Context Variables



4. Type here to search

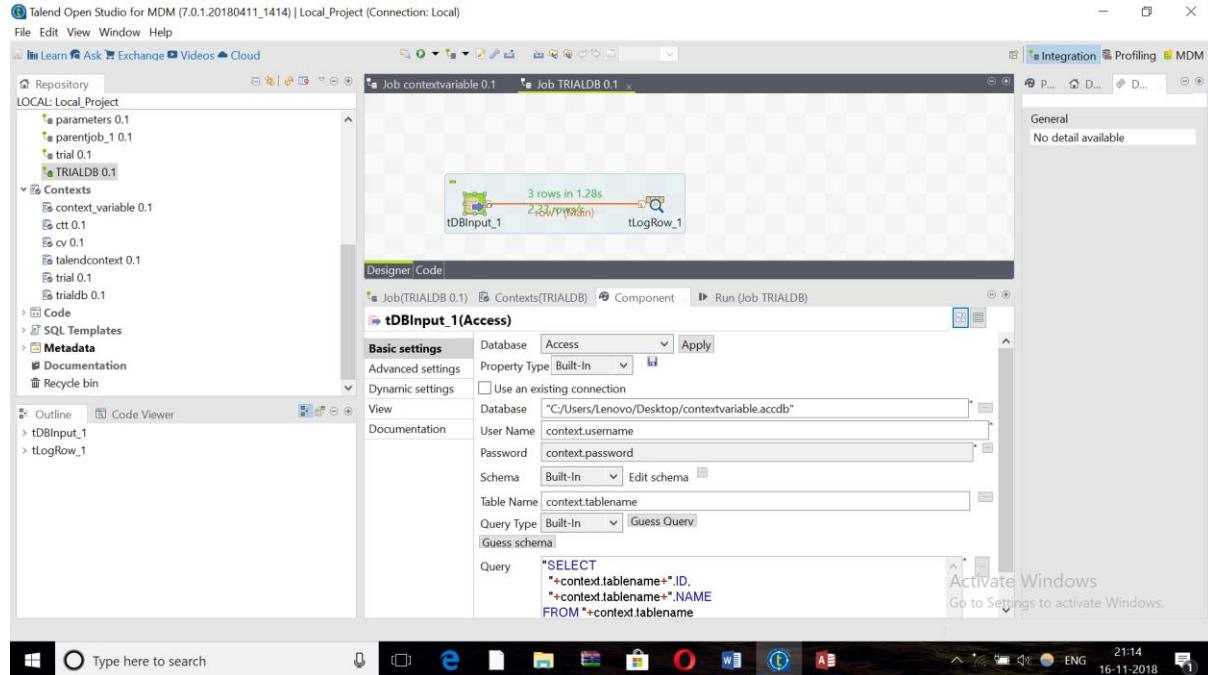
Activate Windows
Go to Settings to activate Windows.

5. Next, we have selected the tdbinput and tlogrow which is connected using row main.

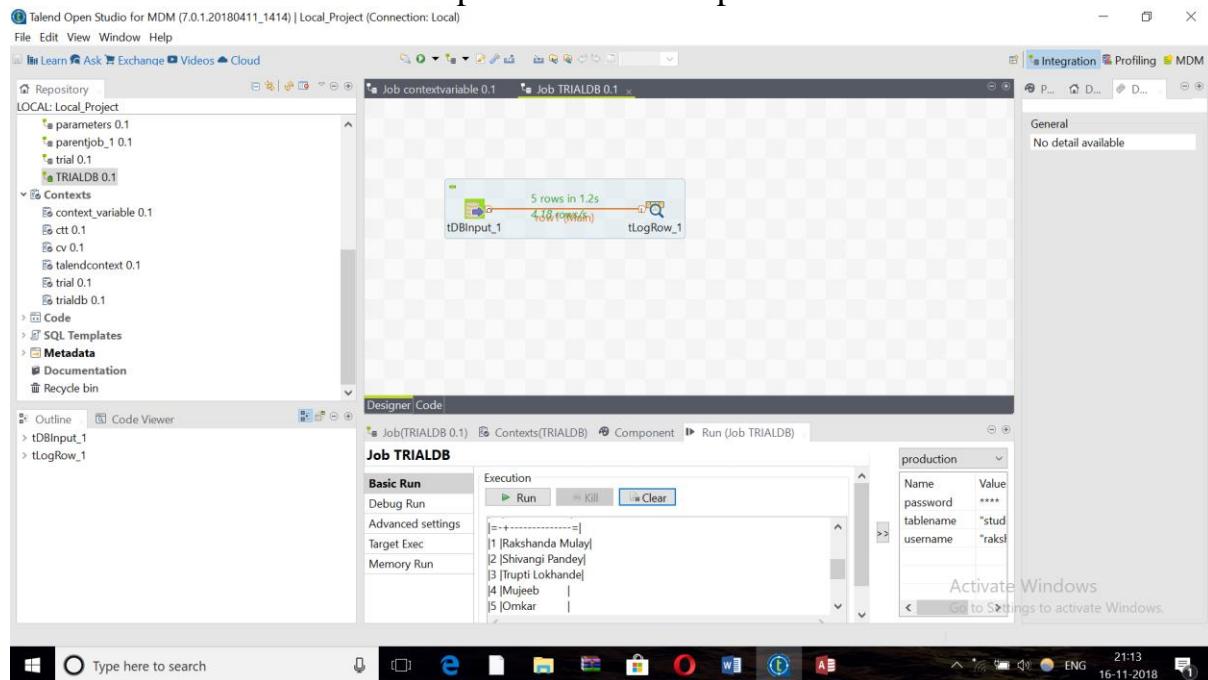


6. In this tdbinput we have to select the access database and select the path of the database saved file. After that in front of username press ctrl+space and then select the context.username variable. In the same way select it for the password and tablename. After that click on guess query and it will automatically display the sql commands.

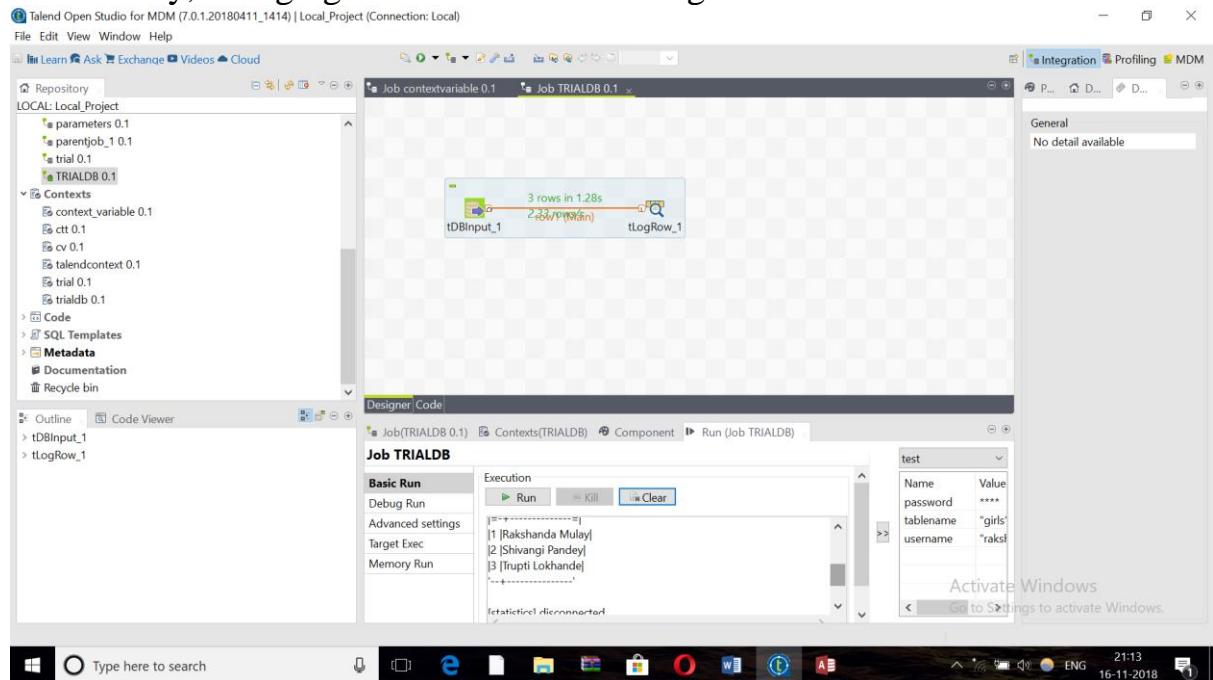
Context Variables



7. Thus run the job in one environment and you will notice that you will get the values which was defined in specific table for specific environment.

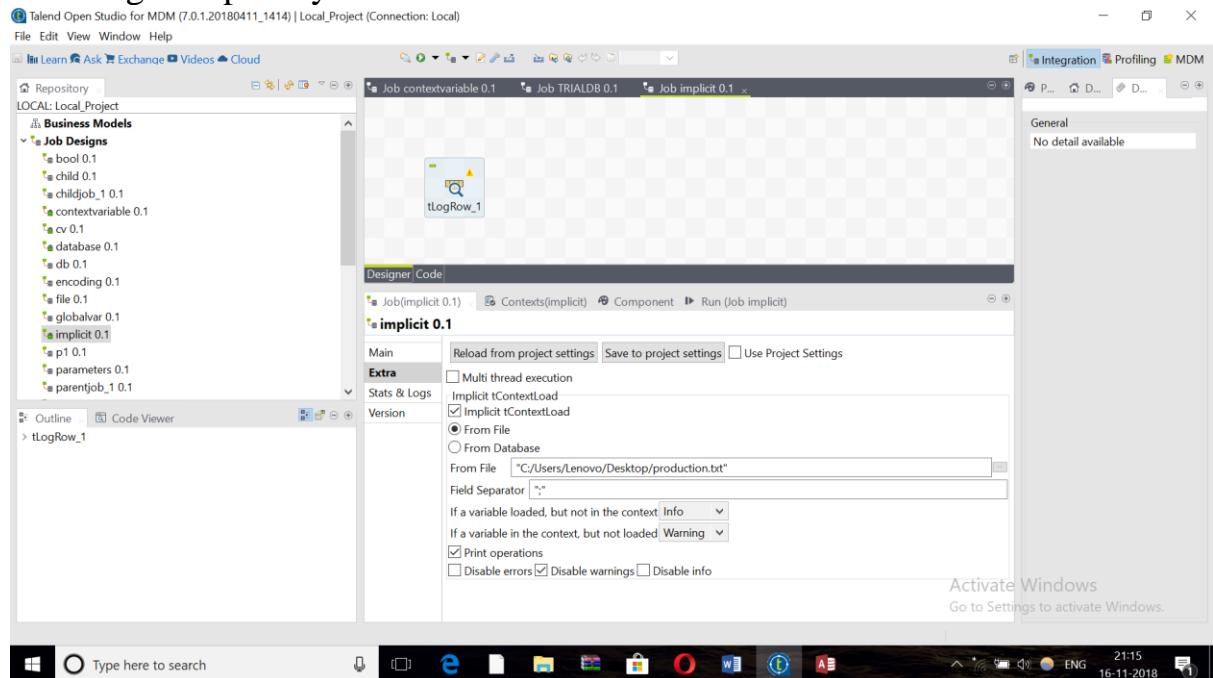


8. Similarly, changing environment will change the table.

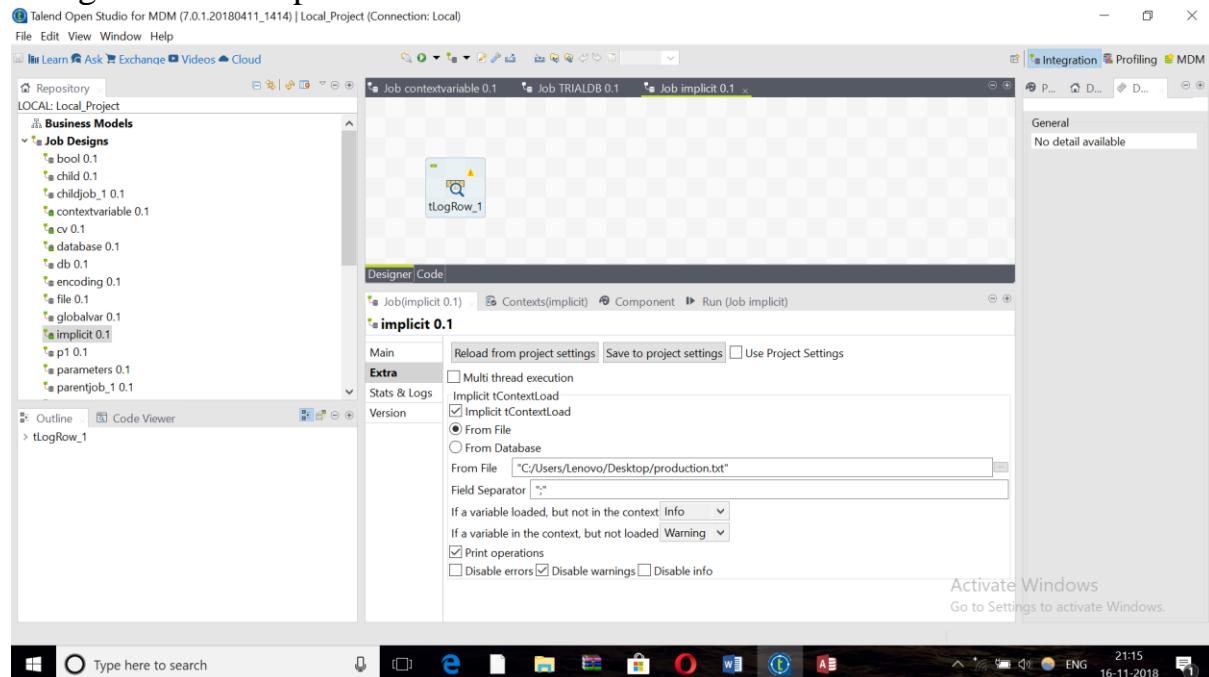


To call file implicitly:

1. Select tlogrow . In the job options go to extra and select the file options with checking it implicitly.



2. Make the file separator with ; and check the print operations. And run the job you will get the values present in the file.

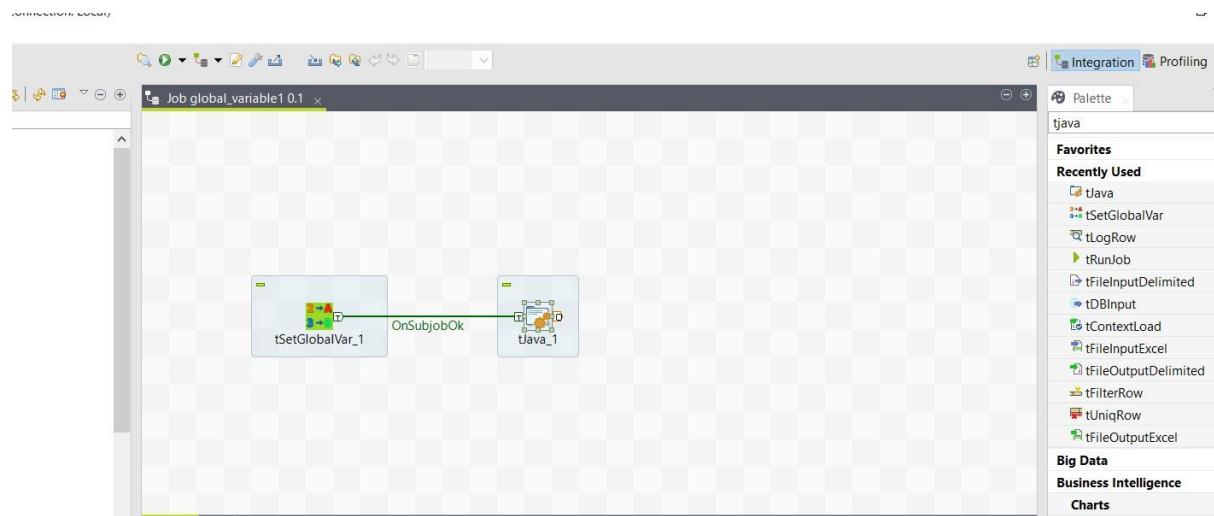


How to Create a Context and Create a globalMap Value Using tJava

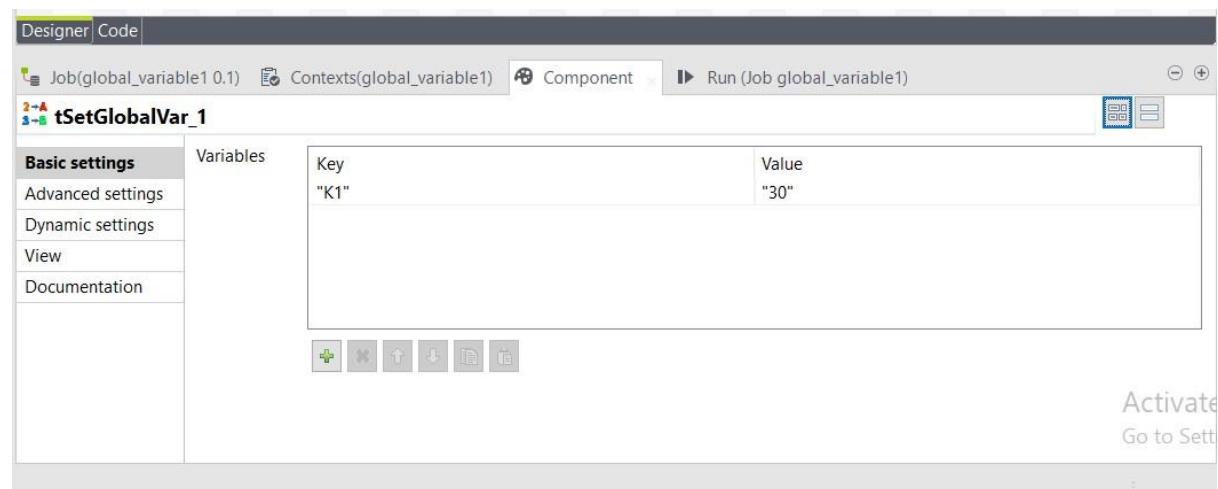
Steps

:

1. Drop tSetGlobalVar and tjava component from palette to design area.
2. Connect both component by using Trigger>On Subjob Ok.



3. Click on tsetglobalVar component and In Basic settings provide key as K1 and Value as 30



4. Click on tJava component and in code

```
area type : String foo = "bar";
```

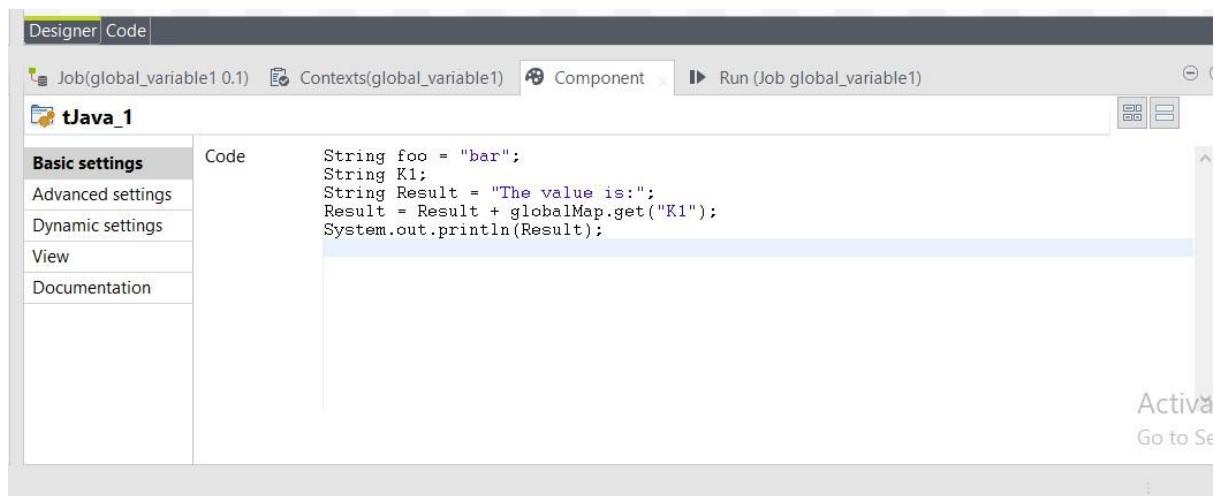
```
String K1;
```

```
String Result = "The value is:";
```

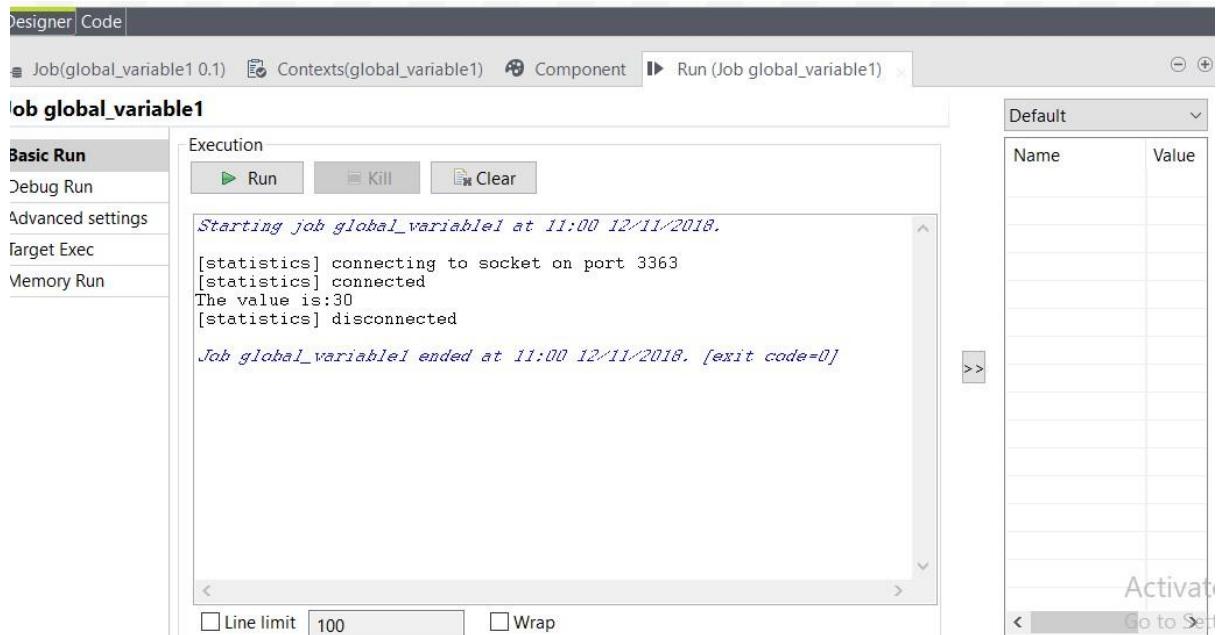
```
Result = Result +
```

```
globalMap.get("K1");
```

```
System.out.println(Result);
```



5. Save and Run the job Value of K1 will be printed on console.

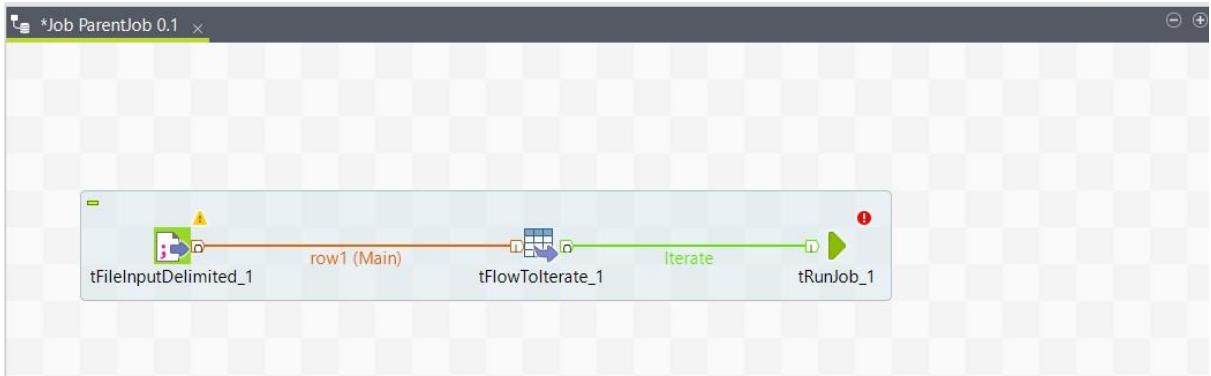


Create a User Defined Global Variable and Pass it as a parameter

- 1) Create a text file as below

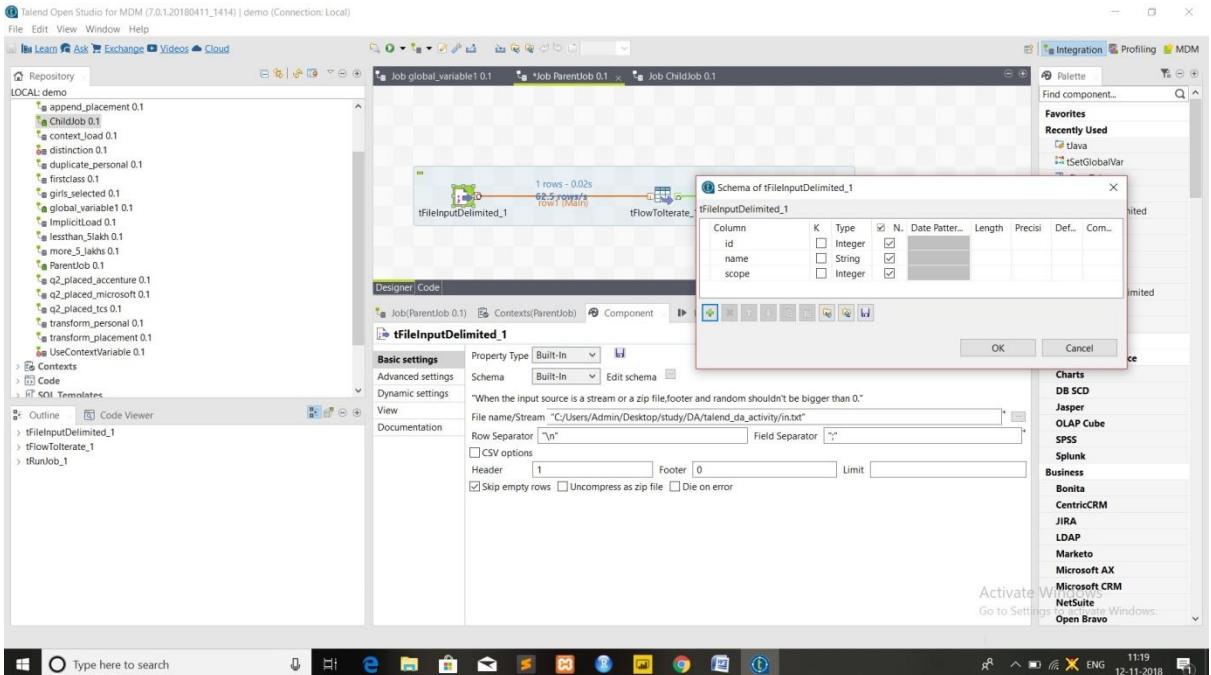
```
id:name:scope
1:Shweta:10
2:Harshali:11
3:Shivangi:12
4:Rakshanda:13
```

- 2) Create a job named as ParentJob

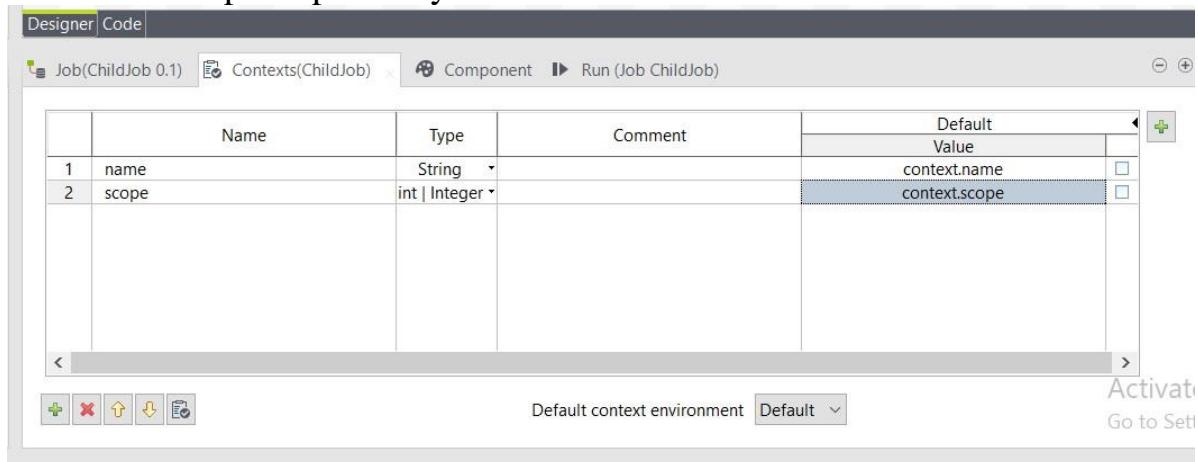


- 3) Drop tfileInputDelimited , tflow to iterate and tRunjob to the workspace
- 4) In the basic settings of tfileInputDelimited, browse for the text file and set the header as 1
Click on edit schema and add three rows. Add the rows as id, name and scope and set the data type as int, string and int respectively.

Context Variables



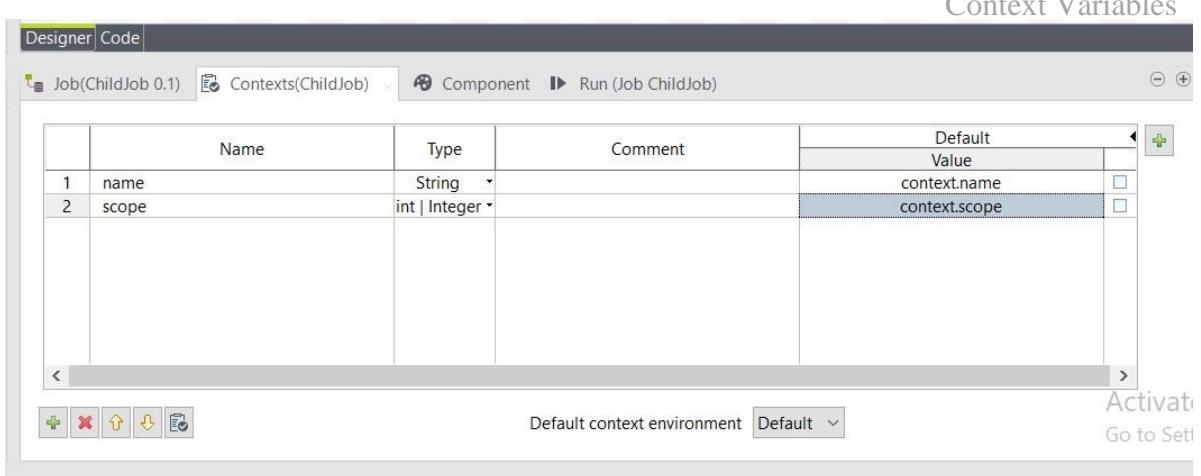
- 5) Click on tFlowIterate and in the basic settings see that the checkbox is checked.
- 6) Now Save the job
- 7) Create a Child Job and in configuration, add two rows name and scope, set the data type as string and integer and in the value as context.name and context.scope respectively.



- 8) Drop tjava to workspace and write the following code
String message="Hi"+context.name+", you have scored "+context.scope+"in the exam";
System.out.println(message);
- 9) Save this job.

- 10) Come back to parentjob and click on the trunjob.

- 11) In the basic settings: Context Param field → click [+] → add two parameters name and scope → in the values add
(String)globalMap("row1.name") and (String)globalMap("row1.scope")



12) save and run the parentjob

