



Connecting the
Data-Driven Enterprise >



Lab Guide **DI Basics**

Version 6.2.1

Copyright 2016 Talend Inc. All rights reserved.

Information in this document is subject to change without notice. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Talend Inc.

Talend Inc.
800 Bridge Parkway, Suite 200
Redwood City, CA 94065
United States
+1 (650) 539 3200

Welcome to Talend Training

Congratulations on choosing a Talend training module. Take a minute to review the following points to help you get the most from your experience.

Technical Difficulty

Instructor-Led

If you are following an instructor-led training (ILT) module, there will be periods for questions at regular intervals. However, if you need an answer in order to proceed with a particular lab, or if you encounter a situation with the software that prevents you from proceeding, don't hesitate to ask the instructor for assistance so it can be resolved quickly.

Self-Paced

If you are following a self-paced, on-demand training (ODT) module, and you need an answer in order to proceed with a particular lab, or you encounter a situation with the software that prevents you from proceeding with the training module, a Talend Support Engineer can provide assistance. Double-click the **Live Expert** icon on your desktop and follow the instructions to be placed in a queue. After a few minutes, a Support Engineer will contact you to determine your issue and help you on your way. Please be considerate of other students and only use this assistance if you are having difficulty with the training experience, not for general questions.

Exploring

Remember that you are interacting with an actual copy of the Talend software, not a simulation. Because of this, you may be tempted to perform tasks beyond the scope of the training module. Be aware that doing so can quickly derail your learning experience, leaving your project in a state that is not readily usable within the tutorial, or consuming your limited lab time before you have a chance to finish. For the best experience, stick to the tutorial steps! If you want to explore, feel free to do so with any time remaining after you've finished the tutorial (but note that you cannot receive assistance from Tech Support during such exploration).

Additional Resources

After completing this module, you may want to refer to the following additional resources to further clarify your understanding and refine and build upon the skills you have acquired:

Talend product documentation (help.talend.com)

Talend Forum (talendforge.org/)

Documentation for the underlying technologies that Talend uses (such as Apache) and third-party applications that complement Talend products (such as MySQL Workbench)

**This page intentionally left blank to ensure new chapters
start on right (odd number) pages.**

S T E P - C O N T E N T

LESSON 1 Getting Started with Talend Studio

Getting Started	14
Lesson Overview	14
Objectives	14
Next Step	14
Starting Talend Studio	15
Overview	15
Open a Talend Studio Project	15
Understand the Talend Studio Main Window	16
Quick tour of Talend Studio tool bars	19
Next	21
Creating a First Job	22
Overview	22
Create Job	22
Add components	24
Connect components	26
Configure components	28
Help	33
Next	35
Running a Job	36
Overview	36
Run the Job	36
Next	38
Wrap-Up	39
Next step	39

LESSON 2 Working with Files

Working with Files	42
Lesson Overview	42
Objectives	42
Before you begin	42
Next Step	42
Reading an Input File	43
Overview	43
Create New Job	43

Add Read File Component	45
Configure	46
Schema	48
Display	50
Warnings	50
Next	52
Transforming Data	53
Overview	53
Add Component	53
Add Output Component	54
Map Rows	55
Build Expression	56
Save	59
Configure Output	59
Next	60
Running the Job	61
Overview	61
Examine input file	61
Run	62
Next	63
Combining Columns	64
Overview	64
Edit Schema	64
Combine Columns	65
Run	65
Next	67
Duplicating a Job	68
Overview	68
Duplicate	68
Open	69
Next	69
Challenges	70
Overview	70
Combine Columns	70
Add a Column	70
Add a Zip Code Column	70
Solutions	71
Overview	71
Combine Columns	71
Add a Column	71
Add a Zip Code Column	71
Run	71
Next	71
Wrap-Up	72

Next step	72
-----------------	----

LESSON 3 Joining Data Sources

Joining Data Sources	74
Lesson Overview	74
Objectives	74
Before You Begin	74
Next Step	74
Creating Metadata	75
Overview	75
Create Metadata	75
Next	78
Creating a Join	79
Overview	79
Duplicate the Job	79
Add Second Source	80
Configure tMap	83
Run Job	86
Next	87
Capturing Join Failures	88
Overview	88
Configure Join Model in tMap	88
Add Rejects Table	89
Log Output	91
Run the Job	94
Next	94
Correcting the Lookup	95
Overview	95
Configure tMap	95
Run the Job	97
Clean up the Job	99
Next	100
Wrap-Up	101
Next step	101

LESSON 4 Filtering Data

Filtering Data	104
Lesson Overview	104
Objectives	104
Next Step	104
Filter output data	105
Overview	105
Duplicate Job	105
Add Filter	105
Next	109

Using tMap for Multiple Filters	110
Overview	110
Duplicate Job	110
Edit Output	111
Edit tMap	118
Run Job	124
Next	125
Wrap-Up	126
Next step	126

LESSON 5 Using Context Variables

Using Context Variables	128
Lesson Overview	128
Objectives	128
Next Step	128
Create Context Variables	129
Overview	129
Create a Variable	129
Create a new Context	131
Use the Context Variable	133
Change Contexts	134
Next	134
Using Repository Context Variables	135
Overview	135
Create Repository Context Variable	135
Add Repository Context Variables to a Job	137
Use Repository Context Variables	138
Next	141
Challenges	142
Overview	142
New Context and Variable	142
Solutions	143
Overview	143
New Context and Variable	143
Next	143
Wrap-Up	144
Next step	144

LESSON 6 Error Handling

Error Handling	146
Lesson Overview	146
Objectives	146
Next Step	146
Handling Errors	147
Overview	147

Duplicate Job	147
Create an error	148
Kill the Job	149
Configure the log level in the execution console	154
Next	155
Raise a Warning	156
Overview	156
Limit the Output	156
Raise a warning	158
Next	161
Wrap-Up	162
Next step	162

LESSON 7 Generic Schemas

Working with Generic Schemas	164
Lesson Overview	164
Objectives	164
Next Step	165
Create a Generic Schema & Generate Data Files	166
Objective	166
Create the Generic Schema	166
Generate Sales data files	169
Create additional subjobs	177
Run the Job	181
Next	182
Create Customers Metadata	183
Overview	183
Create File Metadata	183
Configure File Delimited Metadata	185
Configure Customer Schema	189
Next	191
Create Products Metadata	192
Overview	192
Create File Metadata	192
Configure Product Schema	197
Next	198
Wrap-Up	199
Next step	199

LESSON 8 Working with Databases

Working with Databases	202
Lesson Overview	202
Objectives	202
Before you begin	203
Next Step	203

Creating Database Metadata	204
Overview	204
Create Database Connection Metadata	204
Configure Database Metadata	204
Export as context	205
Next	207
Creating the Customer Table	208
Overview	208
Create new Job	208
Create a Table	209
Populate the Customer Table	213
Verify Results	219
Next	222
Creating the Product Table	223
Overview	223
Duplicate Job	223
Create and Populate the Product Table	223
Verify Results	227
Next	228
Creating the Sales Table	229
Overview	229
Create the Table	229
Modify the Components	229
Duplicate the Subjob	231
Verify Results	232
Next	234
Joining the Data	235
Overview	235
Read the Sales Data	235
Read Customer and Product Data	238
Join the Data	240
Next	247
Finalizing the Job	248
Overview	248
Capture Rejects	248
Perform Calculations	254
Next	258
Wrap-Up	259
Next step	259

LESSON 9 Creating Master Jobs

Creating Master Jobs	262
Lesson Overview	262
Objectives	262
Next Step	262

Creating a Master Job	263
Overview	263
Create a Master Job	263
Overriding Context variables	266
Exporting Master Job	268
Next	272
Challenges	273
Overview	273
Total by Customer	273
Clean Up Temp Files	273
Solutions	274
Overview	274
Total by Customer	274
Clean Up Temp Files	274
Next	274
Wrap-Up	275
Next step	275

LESSON 10 Working with Web Services

Working with Web Services	278
Lesson Overview	278
Objectives	278
Next Step	278
Accessing a Web Service	279
Overview	279
Retrieve information from web service	279
Next	285
Wrap-Up	286
Next step	286

LESSON 11 Running Jobs Standalone

Running Jobs Standalone	288
Lesson Overview	288
Objectives	288
Before You Begin	288
Next Step	288
Building a Job	289
Overview	289
Build Job	289
Extract and Run	292
Next	292
Modifying the Job	293
Overview	293
Create a New Version	293
Modify the new version of the Job	295

Build and Run	296
Next	297
Challenges	298
Overview	298
Display Output	298
Build and run	298
Solutions	299
Overview	299
Display Output	299
Build and run	299
Next	299
Wrap-Up	300
Next step	300

LESSON 12 Best Practices & Documentation

Best Practices Documenting a Job	302
Lesson Overview	302
Objectives	302
Next Step	302
Job Documentation	303
Overview	303
Improve Key Job Properties	303
Add a Note	304
Document a Component	305
View and Generate HTML Documentation	306
Next	308
Wrap-Up	309
Next step	309

APPENDIX Additional Information

Lesson 01 - Getting Started	311
Lesson 02 - Working with Files	311
Lesson 03 - Joining Data Sources	312
Lesson 04 - Filtering Data	312
Lesson 05 - Using Context Variables	312
Lesson 06 - Error Handling	312
Lesson 07 - Generic Schemas	312
Lesson 08 - Working with Databases	312
Lesson 09 - Creating Master Job	312
Lesson 10 - Working with Web Services	313
Lesson 11 - Running Jobs Standalone	313
Lesson 12- Best Practices and Documentation	313



Getting Started with Talend Studio

This chapter discusses the following.

Getting Started	14
Starting Talend Studio	15
Creating a First Job	22
Running a Job	36
Wrap-Up	39

Getting Started

Lesson Overview

During this training, you will be assigned a **Talend Data Fabric** training environment. The purpose of this lesson is to get familiarized with the Talend Studio by covering the essentials.

First, you will start Talend Studio and you will explore the different sections of the Main Window.

Then you will create a first Job by adding components, connecting and configuring them.

This Job is very simple: it will display a "*Hello World*" message in the execution console.

Finally, you will execute the Job in the development environment to test it.

Objectives

After completing this lesson, you will be able to:

Start the Talend Studio

Create a new Job

Find and Add components

Connect and configure components

Find Help on components

Run a Job

Everything has already been setup in your training environment.

Next Step

The first step is to open the existing training project in Talend Studio and explore the interface: [Start Talend Studio](#)

Starting Talend Studio

Overview

Before starting to develop using Talend Studio, you need to get familiarized with the environment. The first step is to start the software and open an existing project. Then you will discover the Talend Studio main window and tool bars.

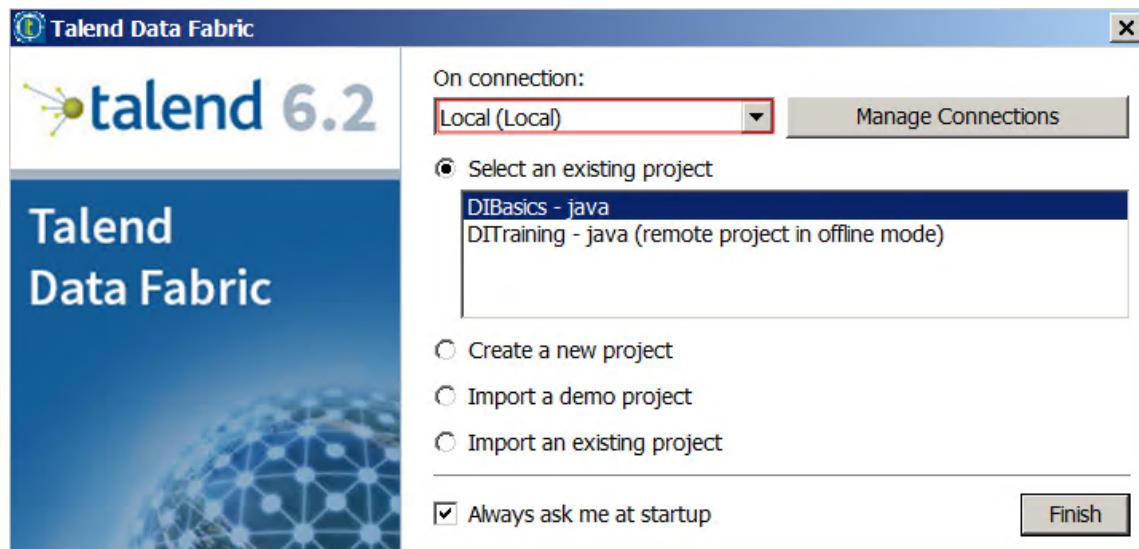
Open a Talend Studio Project



1. Double-click the following icon on your desktop to run Talend Studio:

As an alternative method it's good to know where in the file system the binaries are located so you can start them directly. For example, navigate to C:\Talend\6.2.1\studio. Notice there are many executables, corresponding to various system architectures. The training environment is 64 bit Windows, so double-click on **Talend-Studio-win-x86_64.exe** to invoke the Studio. This is helpful in the event a shortcut does not exist. Also note you can create a shortcut for your desktop here.

2. The **Talend Data Fabric** window opens. Using the **Local** connection, click **Select an existing project** and then select the **DIBasics** project:

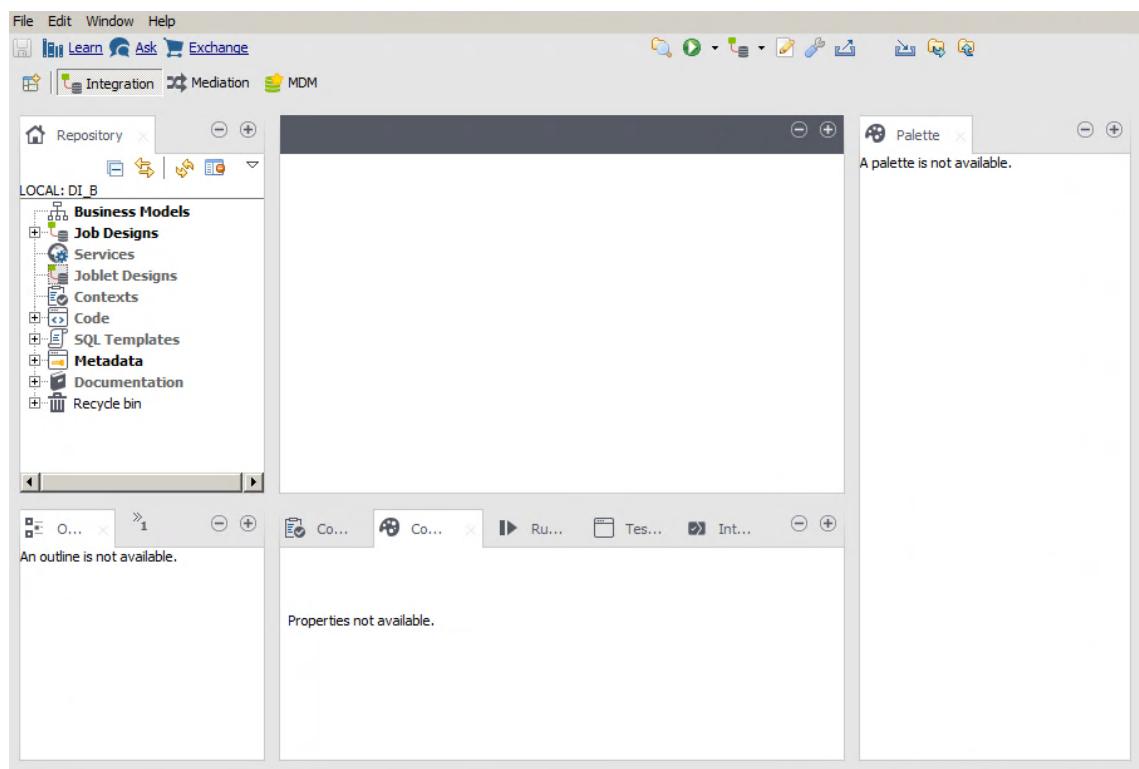


The project has already been created for you.

Note that there are different executable files for different combinations of operating system and CPU. When you are running Talend Software at work or home, choose the appropriate file for your environment. For this training environment the **Talend Data Fabric** is installed.

3. Click **Finish** to open the project. If prompted to join the TalendForge community on your training VM, simply skip the step.

Click **Start now!** if presented with the **Welcome** page. The main project window opens:

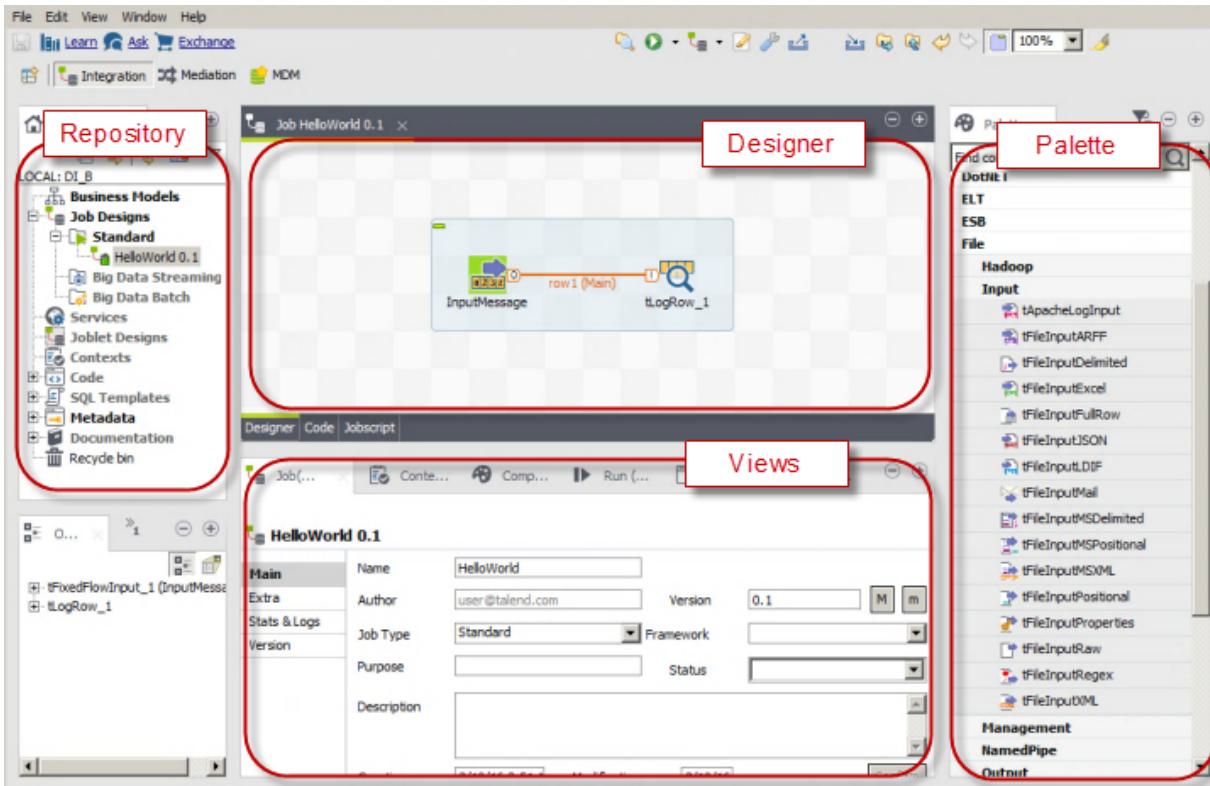


Understand the Talend Studio Main Window

Before starting to build your first Job, let's have a look at the different sections from the **Talend Studio** interface.

For now, the panels and views in your environment are empty as there is no Job defined.

This is how your environment will look after building your first Job in the next section:



We'll use this figure to go through the main panels and views of the Main Window:

- On the left of the window, you can see the **Repository** area containing the project structure with several nodes:

Job Designs contains all your developed Jobs, which also can be organized in folders. A Job Design is a graphical design, of one or more components connected together, that allows you to set up and run data flow management processes. A Job Design translates business needs into code, routines and programs, in other words it technically implements your data flow.

The **Contexts** folder groups files holding the context variables that you want to reuse in various Jobs, such as file paths or database connection details.

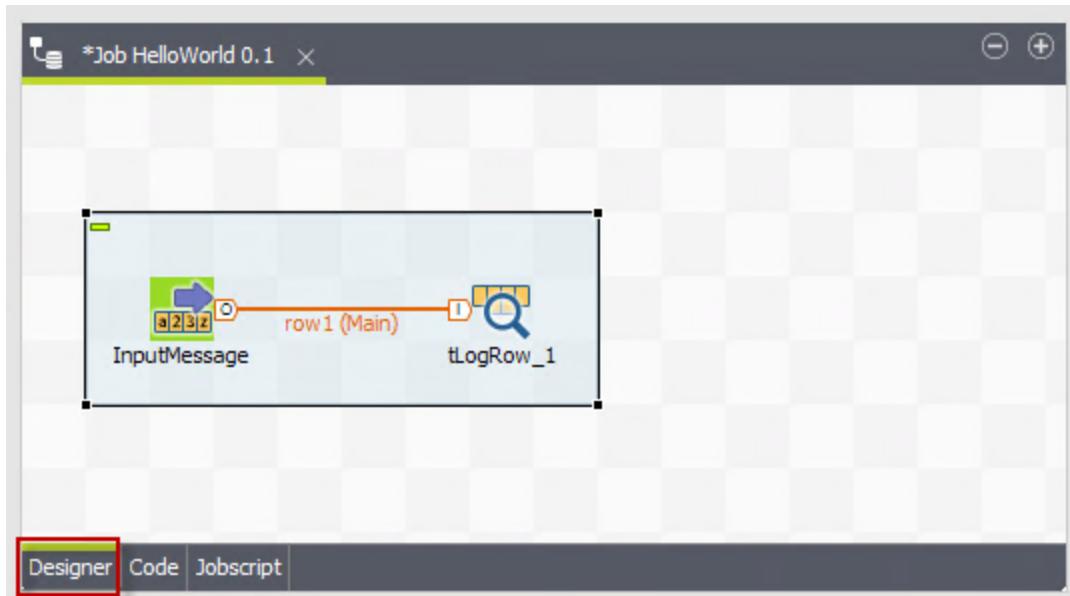
The **Metadata** folder in the Repository tree view stores reusable information on files, databases, and/or systems that you need to access from within your Jobs.
- On the right, you will use the **Palette** to drag and drop technical components to the design workspace in order to design your Job .
- In the middle, have a look at the **Designer** panel. You will use this area to design your Jobs in a graphical mode by adding components and connecting them.

Under the main design tab, you can access several tabs:

Designer tab - It opens by default when creating a Job. It displays the Job in a graphical mode.

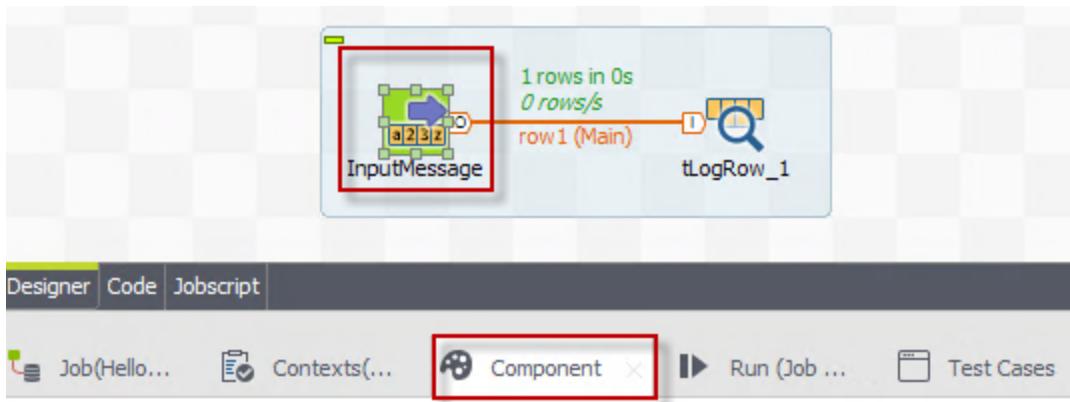
Code tab - It enables you to visualize the code and highlights the possible language errors.

Jobscrip tab - It enables you to visualize and edit the Jobscrip.



4. In the lower half of the design workspace, you can find different **Views**.

The **Component** View displays the properties of the selected element in the design space. These properties can be edited to change or set the parameters related to a particular component.

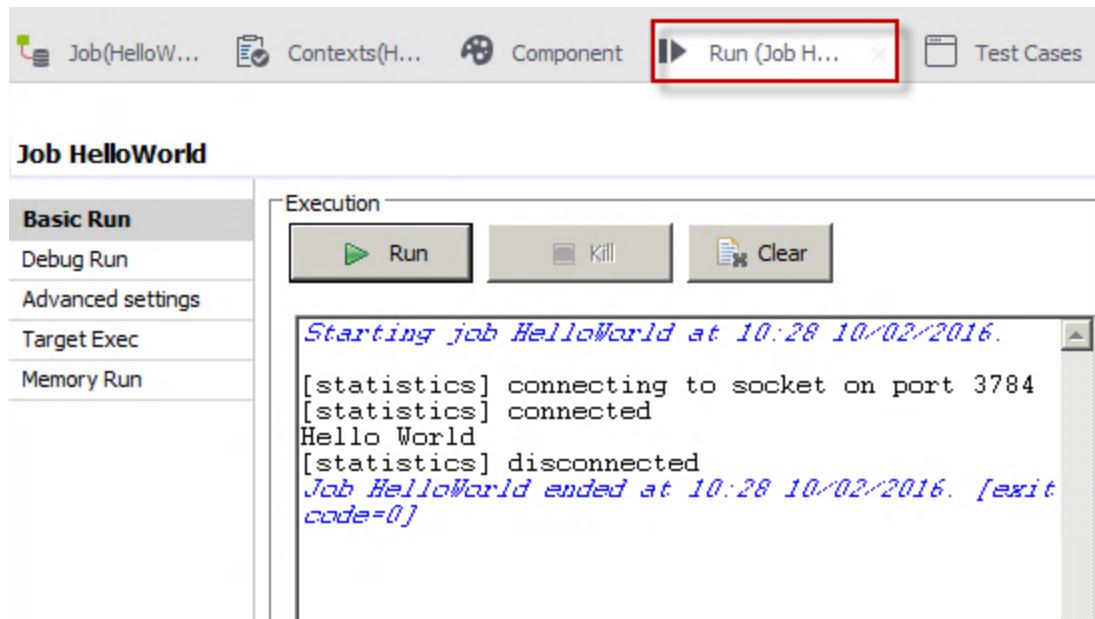


The screenshot shows the 'Component' view for the 'InputMessage' component. The left sidebar lists settings: Basic settings (selected), Advanced settings, Dynamic settings, View, Documentation, and Validation Rules. The main panel shows:

- Schema:** Built-In (dropdown), Edit schema, ...
- Number of rows:** 1
- Mode:** Use Single Table (radio button)
- Values:**

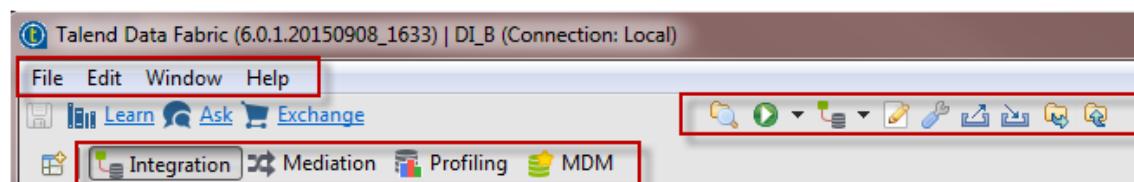
Column	Value
Message	"Hello World"

The **Run** View will be used to execute the Job and explore the results. The results are displayed in the Console in the Execution panel:



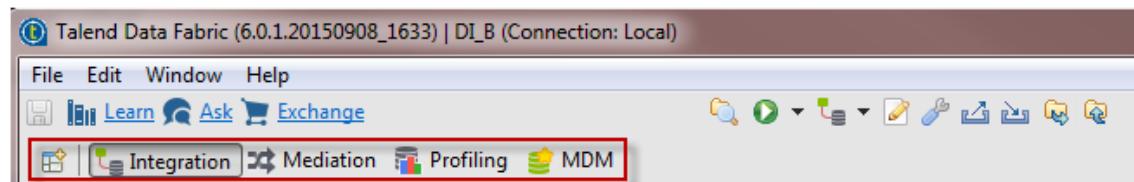
Quick tour of Talend Studio tool bars

- At the top of the **Talend Data Fabric** main window, several tool bars group the most commonly-used Talend features:
 - the main menu
 - a quick access toolbar
 - the perspectives switcher



Depending on the size of your Talend Studio window, the perspectives switcher toolbar will be displayed on the top right corner, just after the quick access toolbar or on the left side, below the quick access toolbar.

- A **Perspective** defines the initial set and layout of views in the Studio. The selected Perspective is always the active one.



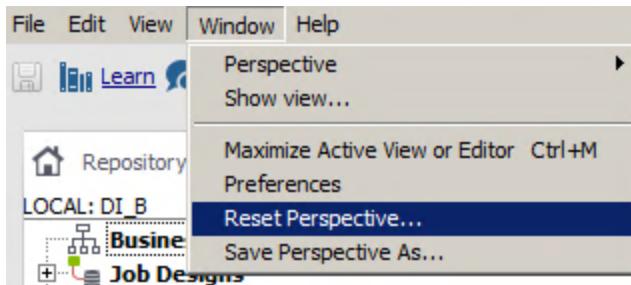
The **Talend Data Fabric** Studio corresponds to the Talend Unified Platform and will be used to create data integration projects but also other projects: Big Data, application integration, master data management or data profiling. That's why several perspectives are available in your training environment.

In this course, only the **Integration** perspective will be used, which is the default perspective when opening the **DIBasics** project.

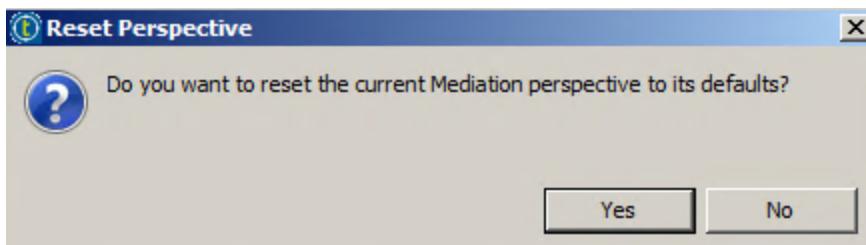
3. If you click and open another perspective, you can always come back to the **Integration** perspective by using one of the following two options:

Option 1: Simply click on the **Integration** button in the toolbar

Option 2: From the menu, go to **Window** and select **Reset Perspective...**



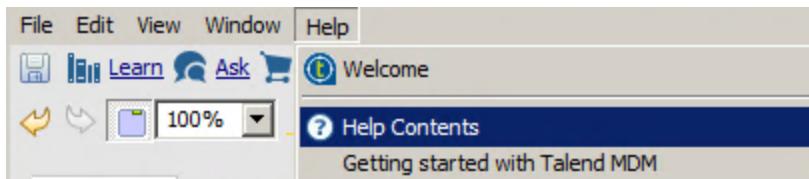
When asked if you want to reset the current perspective to its defaults, answer **Yes**.



The default perspective on your environment is the **Integration** perspective.

Also, by resetting the perspective, the default perspective will open with all the default views in their initial position .

4. From the menu toolbar, choose **Help** and then **Help Contents**



This opens the **Help - Talend Studio** window.

The screenshot shows the 'Help - Talend Studio' window. The left pane is a tree view titled 'Contents' under 'Workbench User Guide'. It includes sections like 'Eclipse platform overview', 'Getting started', 'Basic tutorial', 'The Workbench', 'Editors and views', 'A simple project', 'Closing an editor', 'Navigating resources', 'Files', 'Exporting files', 'Importing files', 'Deleting resources', 'Working with other editors', 'Copying, renaming and moving', 'Searching', 'Tasks and markers', 'Bookmarks', 'Rearranging views and editors', 'Perspectives' (which is selected and highlighted in blue), 'Comparing', and 'Local history'. The right pane displays the content for the 'Perspectives' section, with the URL 'Workbench User Guide > Getting started > Basic tutorial' at the top. The main heading is 'Perspectives'. Below it, a paragraph explains what a perspective is: 'A perspective defines the initial set and layout of views in the Workbench window. One or more perspectives can exist in a single Workbench window.' A sub-section titled 'Perspectives can be opened in one of two ways:' lists two options: 'In the same (existing) Workbench window.' and 'In a new Workbench window.' Another sub-section, 'Perspectives define visible action sets, which can be changed to customize a perspective. A perspective that is built in this manner can be saved, creating a custom perspective that can be opened again later.', follows. At the bottom of the right pane, there is a note: 'The Workbench window displays one or more perspectives. Each product determines initially what default perspective is displayed, in this example it is the Resource perspective. A perspective consists of views such as the Project Explorer as well as editors for working with resources. More than ...'

Please refer to this section to read more about Talend Studio features.

Next

Now that you have opened the training project and you made a quick tour through the Talend Studio interface you can start to build your [First Job](#).

Creating a First Job

Overview

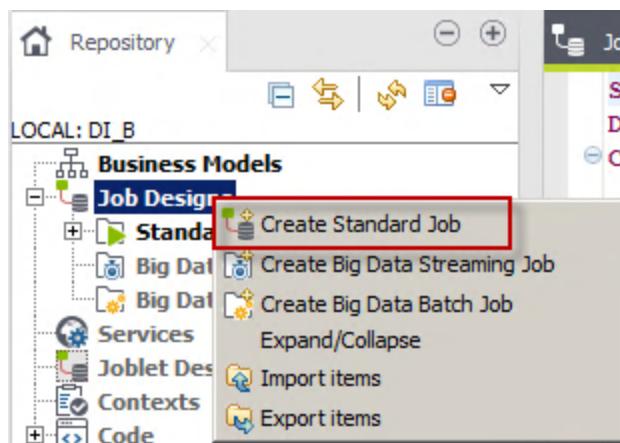
In this section, you will create your first Job and learn how to add and connect components. You will make use of the **Palette** in order to search for components. You will also configure the components in the corresponding **Views** and finally you will execute the Job by using the **Run View**.

The Job is very simple: it will define a "Hello World" message that will be displayed in the execution console.

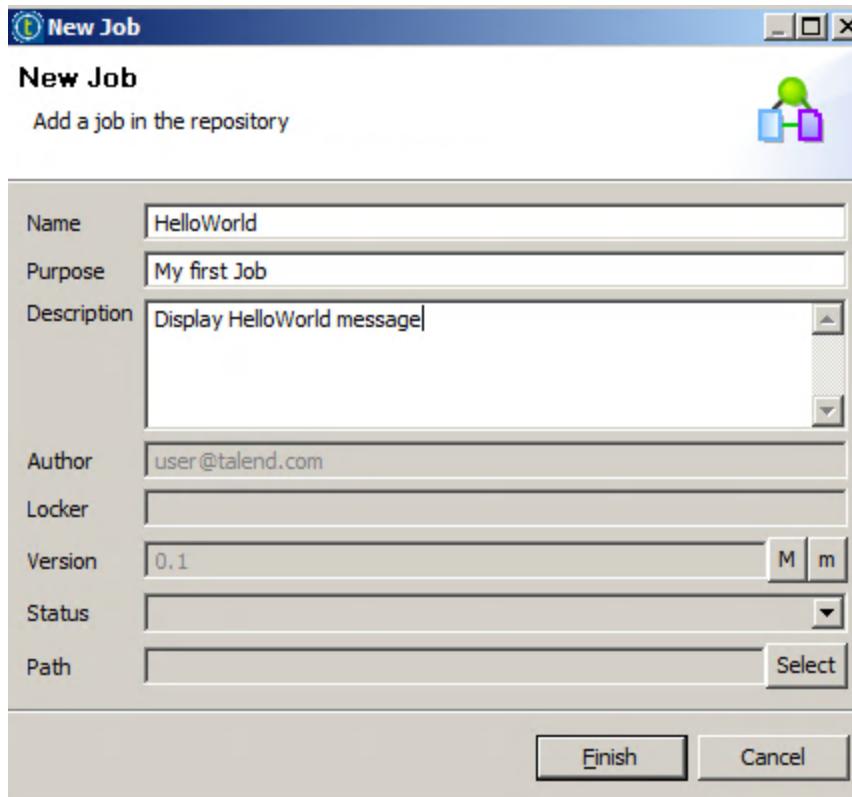


Create Job

1. In order to create a new Job, in the **Repository** section, right-click on **Job Designs** and then click on **Create Standard Job**:



2. Enter *HelloWorld* for the **Name** and then enter the following values for the **Purpose** and the **Description** of your new Job:

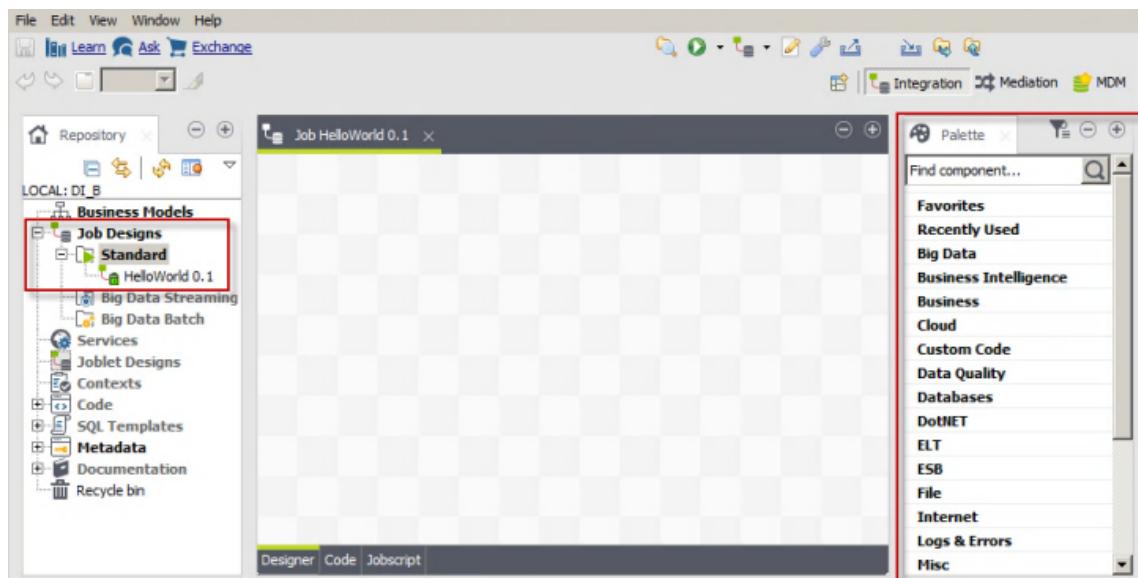


Note that if you try to include illegal characters in the Job name you will be notified in the Studio. For example, "HelloWorld" is legal, but "Hello World" will be flagged due to a space in the Job name. The Studio will issue the following notice: `Name contains incorrect characters.`

3. Click **Finish** when ready.

Notice the changes in the workspace designer, the **Palette**, and the configuration views.

The newly created **Job** is opened in the **Designer** and the components are now available in the **Palette**:

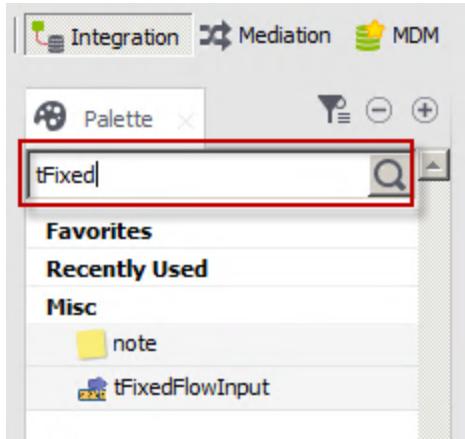


The content of the **Palette** is organized in folders and contains more than 900 components depending on your installation.

Add components

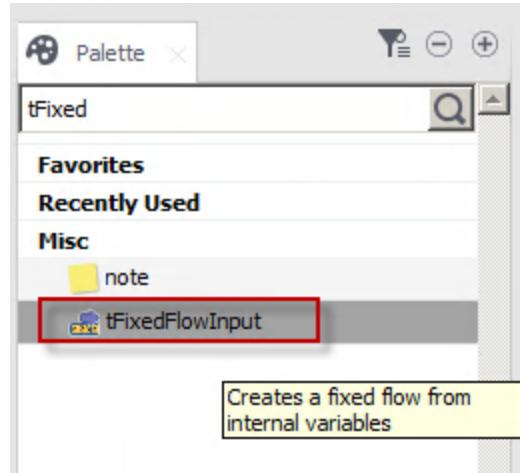
1. Search for the **tFixedFlowInput** component.

Type *tFixed* in the **Find Component** box from the **Palette** and then click the **Search** button nearby or use the **Enter** key from your keyboard:

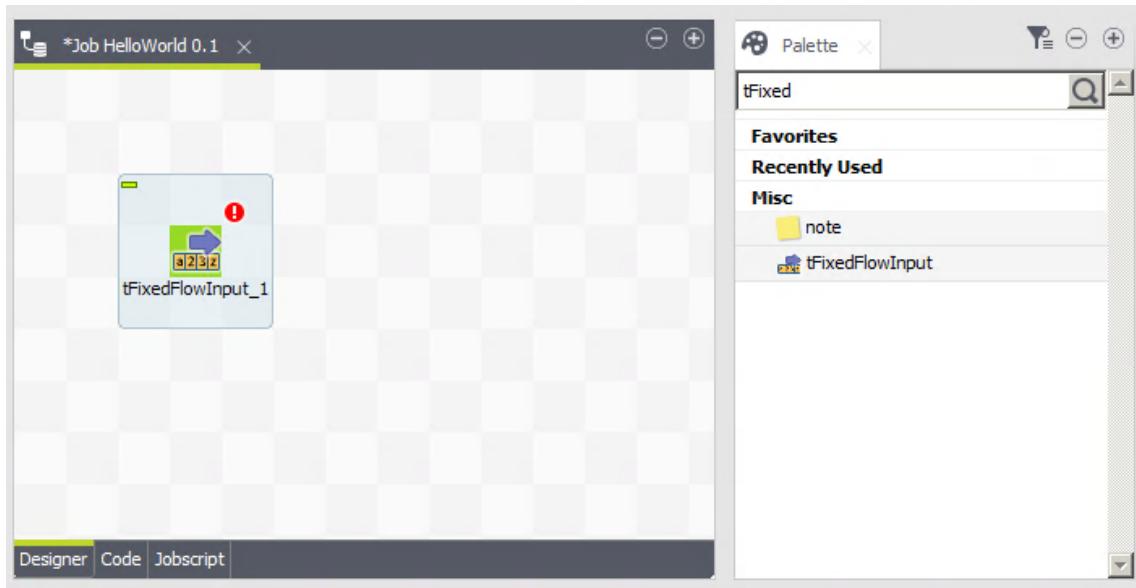


The component has been found under the **Misc** folder.

2. Mouse over the component in the Palette to have a short description of the component. The **tFixedFlowInput** component allows you to define a fixed flow for an internal variable.

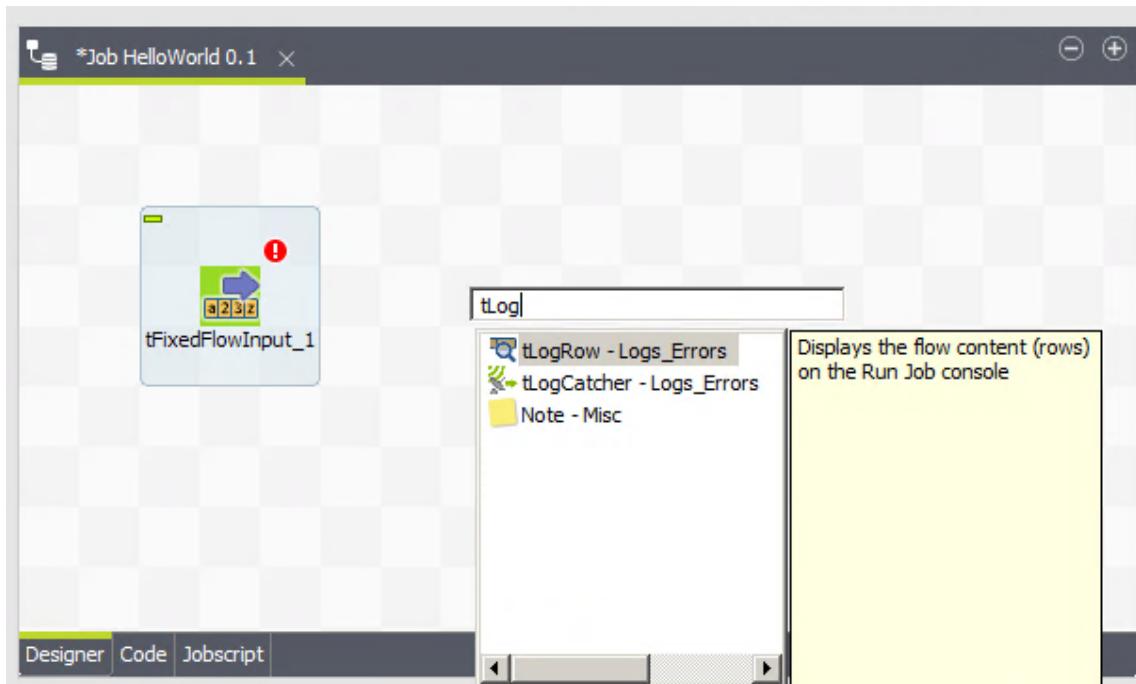


3. Select **tFixedFlowInput** component and with the selected component in the **Palette**, place your mouse in the design space and click. This will add the component to your Job:

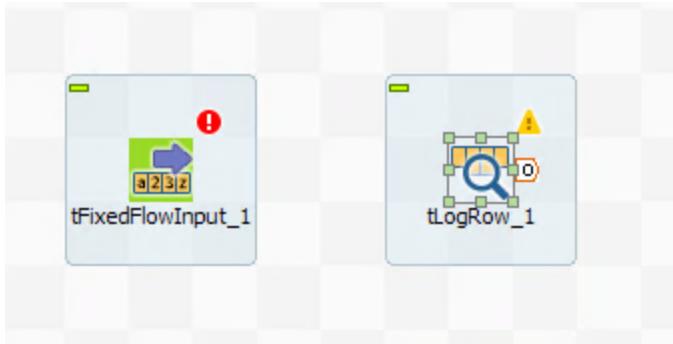


It is also possible to simply drag and drop the component from the **Palette** to your Job's design space.

4. In order to add a second component to your Job, click on the design space outside the component and start typing *tLog*. A list of components will then appear.



5. Double-click the **tLogRow** component.

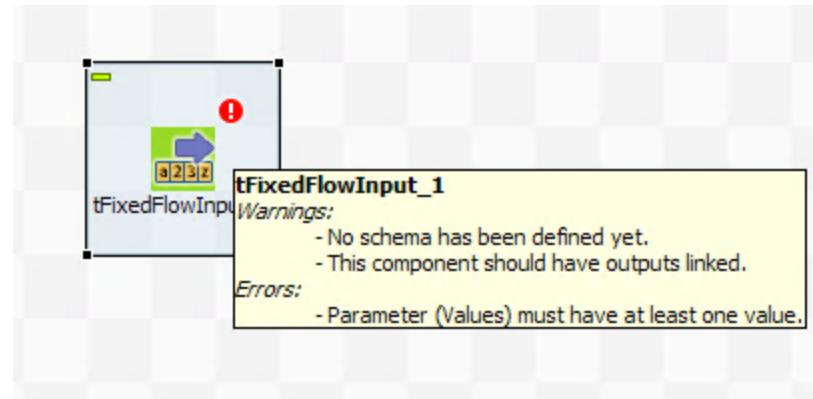


The component is now added to the Job. This is another way you can use to add components to your Job.

As a best practice, arrange your components on the design workspace to mimic the flow of data and execution, from left to right and top to bottom.

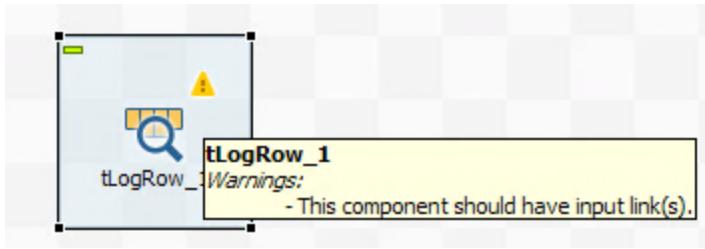
Notice the exclamation marks on the top right corners of the components.

6. Mouse-over the red exclamation mark on the top right corner of the **tFixedFlowInput** component to see the warnings and errors:



The component needs to be configured and has to be connected to an output component.

7. Mouse-over the orange exclamation mark on the top right corner of the **tLogRow** component to see the warnings:

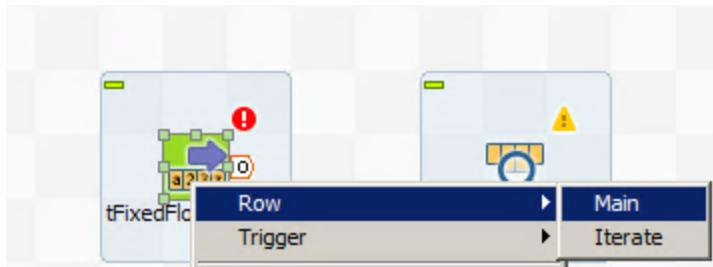


You are now ready to connect the components.

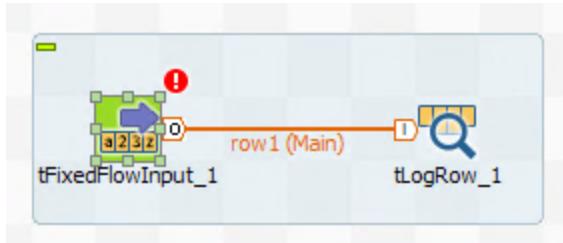
Connect components

1. To link the components, right-click on **tFixedFlowInput** component.

Click Row > Main and then move your mouse over the **tLogRow** component and left-click on it:



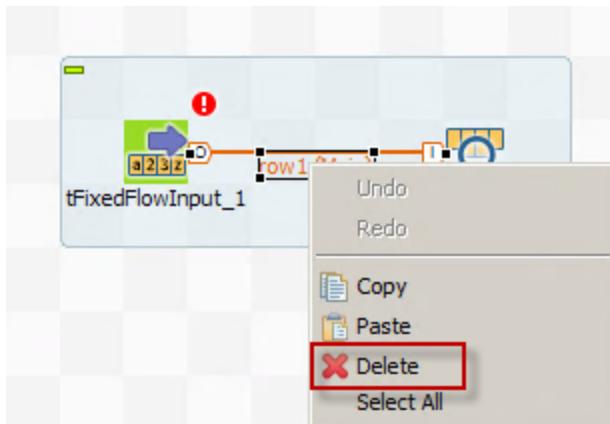
2. The link will be created and displayed:



A row (or link) allows data to flow from one component to another. The arrow indicates the direction of the flow.

In the following steps you will see how you can create the **Main** row differently.

3. Right-click the link and select **Delete**:



Note: It is also possible to use the **Delete** key of the keyboard in order to delete a row or a component.

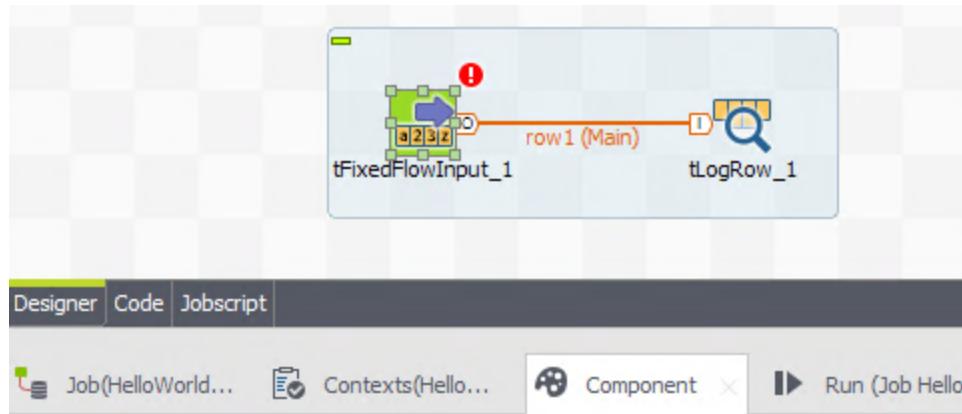
4. In order to re-create the link, right-click the first component, hold the right-click and move the mouse over the second one. This will also create a **Main** link.



While this method requires less operation steps than the previous one, it works only with a limited types of **Row** connections depending on the nature and role of the components you are connecting. More connectors are available in the contextual menu, such as triggers that will be covered later in the course.

Configure components

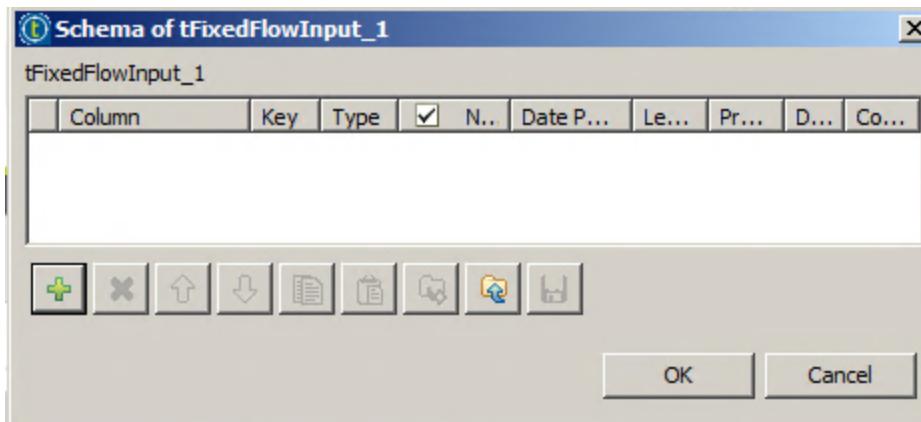
1. Double-click the **tFixedFlowInput** component to open the **Component** view in the lower half of the main window.
Then click on the [...] button just near **Edit schema**.



tFixedFlowInput_1

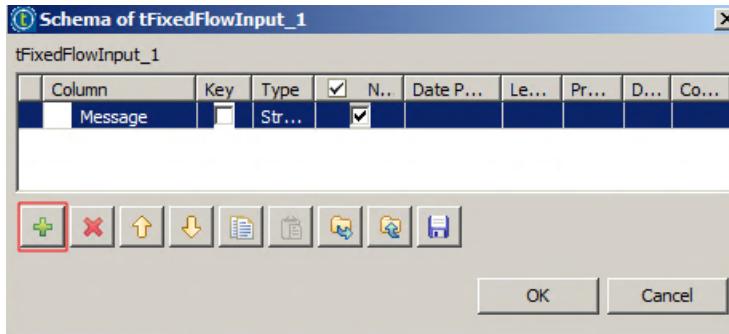
Basic settings	Schema: Built-In <input style="border: 1px solid red; padding: 2px; margin-left: 10px;" type="button" value="Edit schema"/> Number of rows: 1 Mode: <input checked="" type="radio"/> Use Single Table Values: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Column</th> <th style="width: 50%;">Value</th> </tr> </thead> <tbody> <tr> <td colspan="2"></td> </tr> </tbody> </table> <input type="radio"/> Use Inline Table <input type="radio"/> Use Inline Content(delimited file)	Column	Value		
Column	Value				

2. The Schema of tFixedFlowInput wizard opens.

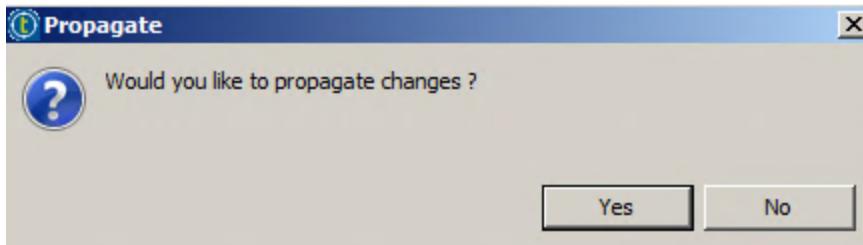


3. Click the [+] green button to add a new column.

Change the Column name to *Message* and then click **OK**:

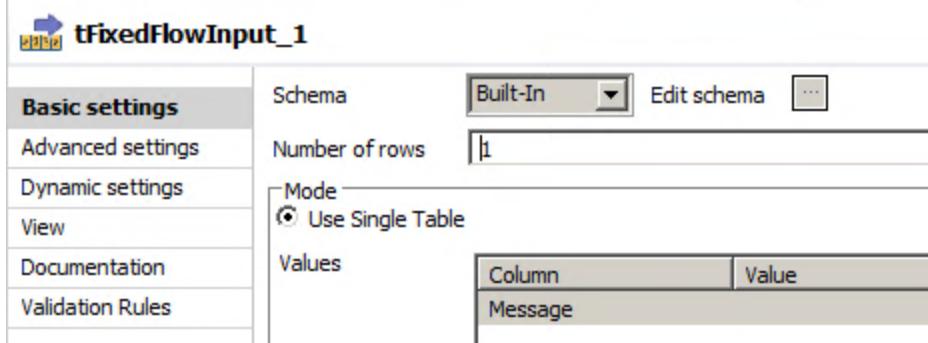


Answer **Yes** when asked if you would like to propagate changes.

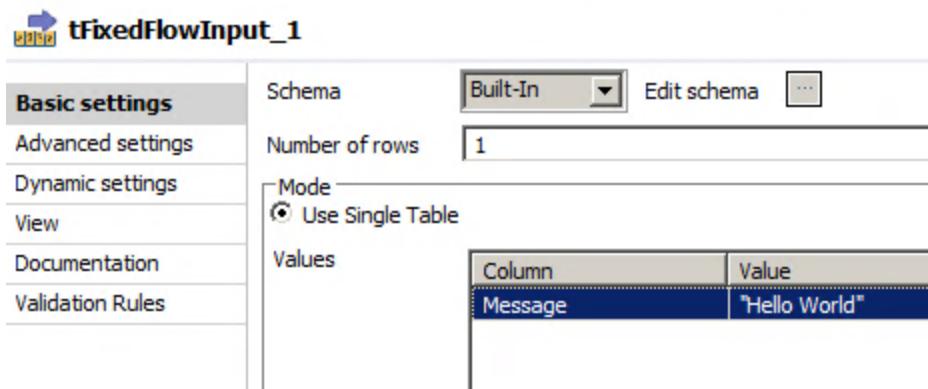


You will get this question every time a schema is updated. In case you have more than one Job using that schema, this one will be propagated to all the Jobs.

- The **Message** column is then added to the schema.



- Set the **Value** of the **Message** Column to "Hello World":



6. Click the **View** section from the **Component** view on the left and then change the **Label format** to *InputMessage*. This updates the component's name in the **Designer**.

The screenshot shows the Talend Designer interface. At the top, there is a toolbar with tabs for 'Designer' (selected), 'Code', and 'Jobscript'. Below the toolbar, the main workspace displays a job flow with two components: 'InputMessage' (a green tFixedFlowInput component) and 'tLogRow_1' (a blue tLogRow component). A connection labeled 'row1 (Main)' links them. In the bottom right corner of the workspace, there is a status bar with icons for 'Job (HelloWorld 0.1)', 'Contexts (HelloWorld)', 'Component', and 'Run (Job Hello...)'.

Below the workspace, a component configuration dialog is open for 'InputMessage(tFixedFlowInput_1)'. The dialog has several sections:

- Basic settings**: Contains fields for 'Label format' (set to 'InputMessage'), 'Hint format' (__UNIQUE_NAME__
__COMMENT__), and 'Connection format' (set to 'row').
- Advanced settings**: Not visible in the screenshot.
- Dynamic settings**: Not visible in the screenshot.
- View**: This section is highlighted with a red box. It contains 'Documentation' and 'Validation Rules' links.

Note that the **tFixedFlowInput** component shows no more errors as the input schema has been defined and the component has an output linked.

7. Now double-click the **tLogRow** component and check the settings in the **Components** view:

The screenshot shows the Talend Data Integration Designer interface. At the top, there are tabs for 'Designer' (selected), 'Code', and 'Jobscrip'. Below the tabs, there are buttons for 'Job (HelloWorld 0.1)', 'Contexts (HelloWorld)', 'Component', and 'Run (Job Hell...)'.

The main area displays a job flow diagram. An 'InputMessage' component is connected to a 'tLogRow_1' component. The 'tLogRow_1' component has a magnifying glass icon and a 'Sync columns' button.

A context menu is open over the 'tLogRow_1' component, showing the following options:

- Basic settings** (selected)
- Advanced settings
- Dynamic settings
- View
- Documentation
- Validation Rules

The 'Basic settings' tab contains the following configuration:

- Schema:** Built-In
- Mode:** Basic (radio button selected)
- Table (print values in cells of a table):** Unselected
- Vertical (each row is a key/value list):** Unselected
- Field Separator:** "|"
- Print header
- Print component unique name in front of each output row
- Print schema column name in front of each value
- Use fixed length for values
- Print content with log4j

Leave the default settings for a first output in the log window.

8. Note there is an * just in front of the name of your Job. This means the Job has not been saved yet. Click the **Save** button and the * disappears:

The screenshot shows the Talend Data Integration Repository interface. The top navigation bar includes 'File', 'Edit', 'View', 'Window', 'Help', 'Learn', 'Ask', and 'Exchange'.

The left sidebar shows a tree structure under 'LOCAL: DI_B' with categories: Business Models, Job Designs, Standard, Services, and Joblet Designs. Under 'Job Designs', 'Standard' is expanded, showing 'HelloWorld 0.1' (marked with a green lock icon).

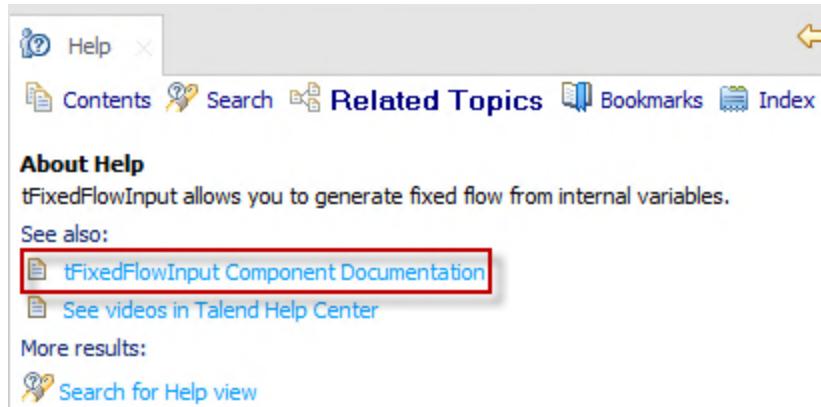
The main workspace shows a job flow diagram with an 'InputMessage' component connected to a 'tLogRow_1' component. The job is titled '*Job HelloWorld 0.1'.

You could also press **Ctrl + S** to save the Job.

9. Your Job is now ready.

Help

1. Click on the **tFixedFlowInput** component (renamed **InputMessage**) and press the **F1** key on your keyboard. This opens the Help page for the selected component:



The screenshot shows a help interface with tabs for Contents, Search, Related Topics (which is selected), Bookmarks, and Index. Below the tabs, there's a section titled 'About Help' with the text: 'tFixedFlowInput allows you to generate fixed flow from internal variables.' Under 'See also:', there are two links: 'tFixedFlowInput Component Documentation' (highlighted with a red box) and 'See videos in Talend Help Center'. A 'More results:' section includes a 'Search for Help view' link.

2. Click on the Component Documentation link highlighted. Then double-click on the **Help** tab to enlarge it so it is more readable:



The screenshot shows a detailed component documentation page for 'tFixedFlowInput'. It includes a navigation bar with 'Prev' and 'Nex' buttons, the title 'tFixedFlowInput', and the chapter 'Chapter 20. Misc group components'. The main content area has two sections: 'Function' (which states 'tFixedFlowInput generates as many lines and columns as you want using the context variables.') and 'Purpose' (which states 'tFixedFlowInput allows you to generate fixed flow from internal variables.').

If you have subscribed to one of the *Talend* solutions with Big Data, this component is available in the following types of Jobs:

- Standard: see [tFixedFlowInput properties](#).
- MapReduce: see [tFixedFlowInput properties in MapReduce Jobs](#).
- Spark Batch: see [tFixedFlowInput properties in Spark Batch Jobs](#).
- Spark Streaming: see [tFixedFlowInput properties in Spark Streaming Jobs](#).

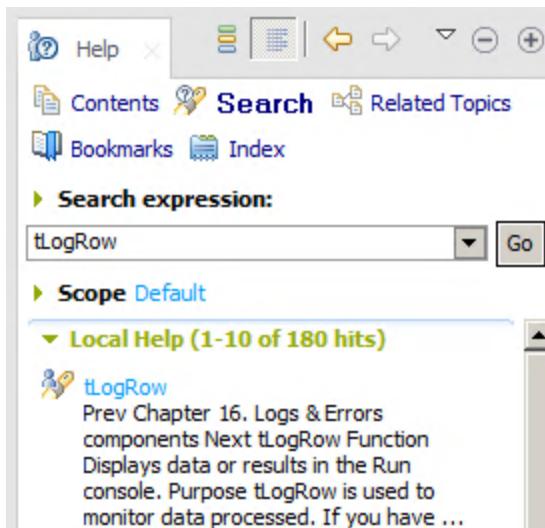
The streaming version of this component is available in the **Palette** of the studio on the condition that you have subscribed to *Talend* Real-time Big Data Platform or *Talend* Data Fabric.

Take a moment to explore the characteristics of the component. Note you can read about different types of Jobs in which the component can be used and you can understand the respective properties. For example, by clicking the **tFixedFlowInput** properties, you will find information about the **Component Family** as well as how to configure the **Schema**:

tFixedFlowInput properties

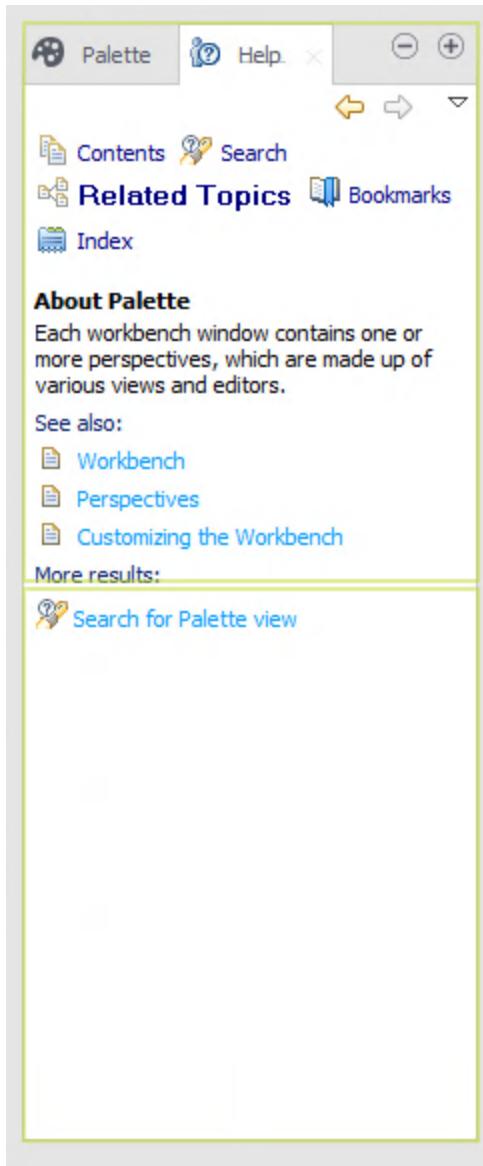
Component family	Misc	
Basic settings	Schema and Edit Schema	<p>A schema is a row description, it defines the number of fields that will be processed and passed on to the next component. The schema is either built-in or remote in the Repository. Click Edit schema to make changes to the schema. If the current schema is of the Repository type, three options are available:</p> <ul style="list-style-type: none"> • View schema: choose this option to view the schema only. • Change to built-in property: choose this option to change the schema to Built-in for local changes. • Update repository connection: choose this option to change the schema stored in the repository and decide whether to propagate the changes to all the Jobs upon completion. If you just want to propagate the changes to the current Job, you can select No upon completion and choose this schema metadata again in the [Repository Content] window.
		Built-in: The schema will be created and stored locally for this component only. Related topic: see <i>Talend Studio User Guide</i> .

3. Then click on the **Search** button and find the documentation for the **tLogRow** component.



4. Double-click the **Help** tab to reset back the size of the page.

You could also drag and drop it to any section of the Main Window.



Use the same method to move other tabs and organize your workspace as you prefer.

Recall that if you want to get back to the initial organization of the workspace, you can use the **Window > Reset Perspective...** option in the main menu.

Next

You have now created your first Talend Data Integration Job. It's now time to [Run it](#).

Running a Job

Overview

Run the Job

1. To run the Job, open the **Run** view (just near the **Component** tab in the lower half of the window):

The screenshot shows the Talend Studio interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, etc. Below the toolbar, the title bar says "Job HelloWorld 0.1". The main workspace is a grid-based canvas where a job flow is being built. A specific step in the flow is highlighted with a blue rounded rectangle, showing a green arrow pointing from one component to another. Below the canvas, there's a navigation bar with tabs: "Designer" (which is selected and highlighted in yellow), "Code", and "Jobscript". In the bottom left corner of the workspace, there's a small preview of the job flow. At the bottom of the screen, there's a dock with several open views: "Job (HelloWorld 0.1)", "Run (Job HelloWorld...)" (which is currently active), and "Contexts(HelloWo...)".

Job HelloWorld

Basic Run

- Debug Run
- Advanced settings
- Target Exec
- Memory Run

Execution

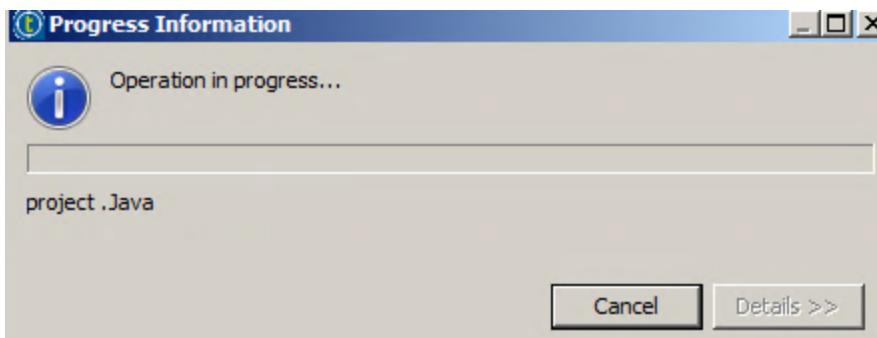
Run (highlighted with a red box)

Kill

Clear

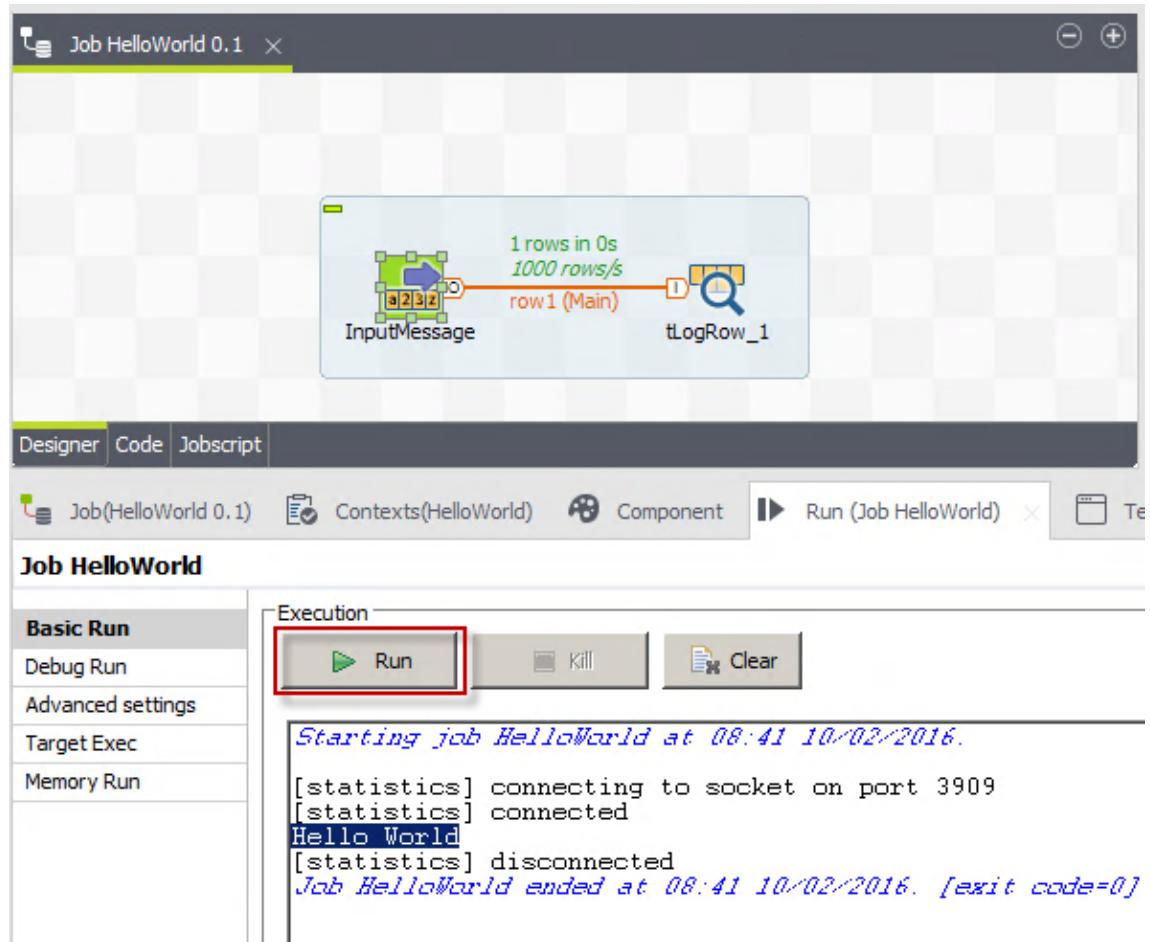
This is where you execute a Job.

2. Click the **Run** button and wait for the Job to execute:



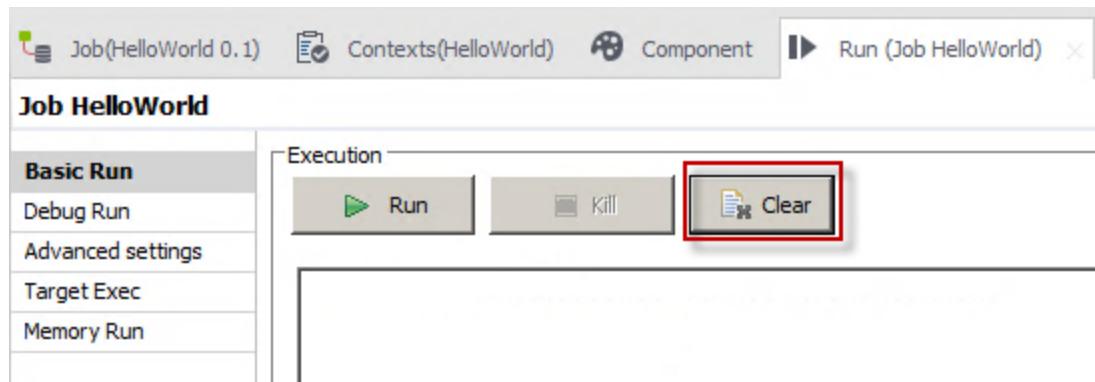
3. Talend Studio saves the Job, builds it, and then runs it.

The Job finishes quickly, displaying messages in the **Run** view:

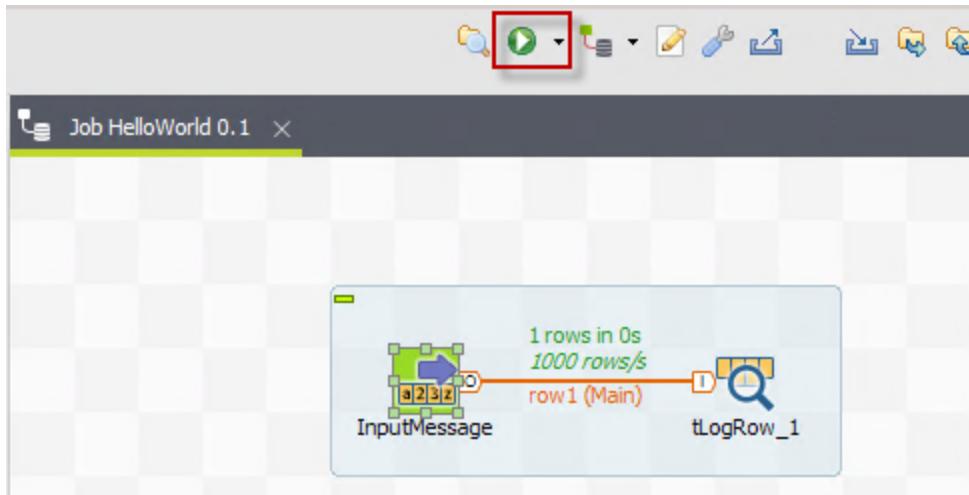


You can see the *Hello World* message is then displayed in the execution console.

4. Click the **Clear** button to clear the execution console.



5. As an alternative, in order to run your Job, you can also use the **Run** icon above the design workspace:



Next

Congratulations! You have now built and run your first Talend Data Integration Job.

It's now time to [Wrap-Up](#).

Wrap-Up

In this lesson, you covered the basics required to build data integration Jobs in the Talend Studio.

You started the **Talend Data Fabric** Studio and you explored the different sections from the workbench, including the **Repository**, the Design workspace, the **Palette** and the configuration views.

Then you built your first Job that used several components. You experienced different methods to add and connect the components and you learned how to configure them. You used the **Palette** to search for the components and you opened the component's documentation from the **Help** page.

While building your first Job you learned that **tFixedFlowInput** component is used to assign fixed flow for internal variables and that the **tLogRow** component is used to monitor data processed.

You will have the chance to use other components and create other Jobs in the following lessons.

Next step

Congratulations! You have successfully completed this lesson. To save your progress, click **Check your status with this unit** below. To go to the next lesson, on the next screen, click **Completed. Let's continue >**.

**This page intentionally left blank to ensure new chapters
start on right (odd number) pages.**