



Talend Data Services Platform

Getting Started Guide

6.1.1

Adapted for v6.1.1. Supersedes previous releases.

Publication date: December 10, 2015

Copyright © 2015 Talend Inc. All rights reserved.

Notices

Talend, Talend Integration Factory, Talend Service Factory, and Talend ESB are trademarks of Talend, Inc.

Apache CXF, CXF, Apache Karaf, Karaf, Apache Camel, Camel, Apache Maven, Maven, Apache Syncope, Syncope, Apache ActiveMQ, ActiveMQ are trademarks of The Apache Foundation. Eclipse Equinox is a trademark of the Eclipse Foundation, Inc. SoapUI is a trademark of SmartBear Software. Hyperic is a trademark of VMware, Inc. Nagios is a trademark of Nagios Enterprises, LLC.

All other brands, product names, company names, trademarks and service marks are the properties of their respective owners.

End User License Agreement

The software described in this documentation is provided under **Talend's** End User License Agreement (EULA) for commercial products. By using the software, you are considered to have fully understood and unconditionally accepted all the terms and conditions of the EULA.

To read the EULA now, visit <http://www.talend.com/legal-terms/us-eula>.

Table of Contents

Preface	v
1. General information	v
1.1. Purpose	v
1.2. Audience	v
1.3. Typographical conventions	v
2. Feedback and Support	v
Chapter 1. Getting Started with Talend Studio	1
1.1. Launching Talend Studio	2
1.1.1. How to launch the Studio for the first time	2
1.1.2. How to connect to TalendForge	4
1.1.3. How to access a Repository	6
1.1.4. How to set up a project in the repository	9
1.2. Working with different workspace directories	10
1.2.1. How to create a new workspace directory	10
1.2.2. How to connect to a different workspace directory	11
1.3. Working with projects	13
1.3.1. How to create a project	13
1.3.2. How to create a sandbox project	14
1.3.3. How to import a demo project	16
1.3.4. How to open a local project	18
1.3.5. How to open a remote project	18
1.4. Managing licenses	20
1.4.1. Setting a license for the Studio	20
1.4.2. Checking and replacing the license for the Studio	21
1.5. Multi-perspective approach	22
1.5.1. Switching between different perspectives	22
1.5.2. Saving the configuration of a perspective	25
1.5.3. Managing quick access icons	26
Chapter 2. Working in Talend Studio - basic data integration Job examples	27
2.1. Getting started with a basic Job	28
2.1.1. Creating a Job	28
2.1.2. Adding components to the Job	29
2.1.3. Connecting the components together	32
2.1.4. Configuring the components	33
2.1.5. Executing the Job	35
2.2. Use cases	36
2.2.1. Updating data in a database table	36
2.2.2. Mapping data using a filter and a simple explicit join	39
Chapter 3. Working in Talend Studio - basic Service and Route examples	45
3.1. Building a simple SayHello data service	46
3.1.1. Creating a SayHello provider	46
3.1.2. SayHello consumer	52
3.2. SayHelloRoute example	55
3.2.1. Creating the route	56
3.2.2. Running the services	59
Chapter 4. Profiling, cleansing and monitoring data	63
4.1. Profiling customer data	64
4.1.1. Identifying data anomalies	64
4.1.2. Sharing analysis results: reports	74
4.2. Cleansing data	79
4.2.1. Removing duplicate values	79
4.2.2. Removing non-matching values	80
Appendix A. Glossary	83

Preface

1. General information

1.1. Purpose

This guide aims at helping users get started with the *Talend Data Services Platform* quickly. For detailed explanations on features and functions of the *Talend Data Services Platform*, see the other documentation delivered with the *Talend Data Services Platform*.

Information presented in this document applies to *Talend Data Services Platform 6.1.1*.

1.2. Audience



This guide is for users and administrators of *Talend Data Services Platform*.



The layout of GUI screens provided in this document may vary slightly from your actual GUI.

1.3. Typographical conventions

This guide uses the following typographical conventions:

- text in **bold**: window and dialog box buttons and fields, keyboard keys, menus, and menu and options,
- text in **[bold]**: window, wizard, and dialog box titles,
- text in `courier`: system parameters typed in by the user,
- text in *italics*: file, schema, column, row, and variable names,
- text in *italics*: file, schema, column, row, and variable names,
- The  icon indicates an item that provides additional information about an important point. It is also used to add comments related to a table or a figure,
- The  icon indicates a message that gives information about the execution requirements or recommendation type. It is also used to refer to situations or information the end-user needs to be aware of or pay special attention to.
- Any command is highlighted with a grey background or code typeface.

2. Feedback and Support

Your feedback is valuable. Do not hesitate to give your input, make suggestions or requests regarding this documentation or product and find support from the **Talend** team, on **Talend's** Forum website at:

<http://talendforge.org/forum>



Chapter 1. Getting Started with Talend Studio

This chapter provides basic information required to get started with *Talend Studio*, including launching *Talend Studio* and creating projects.

After this chapter, you are encouraged to try your own Job designs in your *Talend Studio* by following the examples given in this guide, and read the *Talend Studio User Guide* to learn more about your *Talend Studio*.

1.1. Launching Talend Studio

This section guides you through the basics for launching *Talend Studio* for the first time, connecting to a local or remote repository, and opening your first project in the Studio, and provides information on setting up a project.

1.1.1. How to launch the Studio for the first time

To open *Talend Studio* for the first time, complete the following:

1. Uncompress the *Talend Studio* zip file and, in the folder, double-click the executable file corresponding to your operating system.



The Studio zip archive contains binaries for several platforms including Mac OS X and Linux/Unix.

2. In the **[User License Agreement]** dialog box that opens, read and accept the terms of the end user license agreement to proceed.
3. In the dialog box that opens prompting you to load your product license, select an option to specify your license, before clicking **Next** to load it into your Studio.
 - If you have already set your license and project in *Talend Administration Center*, select **My product license is on a remote host**, fill in the credentials and the URL of your *Talend Administration Center* Web application, and click **Fetch** to retrieve the license. In this way, you do not have to set up a remote repository as the settings of the project you created in the Web application are automatically retrieved. Then, click **Next** to go to the login window to:
 - open an existing remote project as detailed in [How to open a remote project](#), or
 - create a new sandbox project as detailed in [How to create a sandbox project](#).
 - If your license file is stored locally on your computer, select **My product license is on the local file system**, click **Browse** to browse to your license file, and then double-click it.

Alternatively you can quickly load your license by dragging and dropping your locally stored license file directly onto the right panel of this dialog box without having to select an option.

Please import your product license from Administration Center or browse your local file system for it:

☐ My product license is on a remote host:

Login:

Password:

Server URL:

☒ My product license is on the local file system:

This procedure assumes the your license file is stored locally, so select the second option and browse to and open your license file, and then click **Next** to load your license and go to the next step to set up your project.



The license comes with your *Talend Studio* product subscription and determines the edition of *Talend Studio* you can have access to.

4. In the *Talend Studio* login window, select an option to define your project that will hold all Jobs and Business models designed in the Studio.



This login window appears only when the Studio is started for the first time with a license loaded from your local file system. When you launch the Studio again using a locally loaded license, the normal login window opens, which provides one more option, a connection list box, for subscription-based users to select a repository connection when launching the Studio.

If you plan to use the same repository connection and / or project at your next Studio launch, you can skip the login window to speed up Studio launch by clearing the **Always ask me at startup** check box. Then, if you want to see the login window again, go to the menu **Window > Preferences** to open the [Preferences] window, select **Talend**, and select the **Always show project dialog at startup** check box.

- Select **Create a new project**, specify a project name and click **Finish** to create a new local project. For more information, see [How to create a project](#).
- Select **Import a demo project** and click **Finish** to import a demo project that includes numerous samples of ready-to-use Jobs. This Demo project can help you understand the functionalities of different *Talend* components. For more information, see [How to import a demo project](#).
- Select **Import an existing project** and click **Finish** to import an existing projects stored locally. For more information, see your *Talend Studio User Guide*.
- If you want to modify the default repository connection or create a new one, click **Manage Connections** to set up your connection before setting up a project. For further information about connecting to a repository, see [How to access a Repository](#).

As the purpose of this procedure is to create a new localproject, select **Create a new project**, fill in a project name in the text field, and click **Finish**.

5. Depending on the license you are using, you will see either of the following:
 - A Quick Tour to *Talend Studio*. Click **Next** to go the next slide of the presentation, or click **Close** to end the presentation and display the main window of your *Talend Studio*.

This presentation automatically starts at the Studio initial launch. To open it manually later, go to **Help > Studio Quick Tour** from the Studio menu bar.

- The **[Welcome]** window, which provides direct links to Demo projects, user documentation, tutorials, **Talend** forum, **Talend** on-demand training and **Talend** latest news. Click **Start now!** to open *Talend Studio* main window, which displays a page that provides useful tips for beginners on how to get started with the Studio. Clicking an underlined link brings you to the corresponding tab view or opens the corresponding dialog box.

When the **[Additional Talend Packages]** wizard opens, install additional packages such as language packs if needed. For more information, see the section about installing additional packages in the *Talend Installation Guide*.

6. You can skip this installation step and close the wizard by clicking **Cancel**.

This wizard appears each time you launch the studio if any additional package is available for installation unless you select the **Do not show this again** check box. You can also display this wizard by selecting **Help > Install Additional Packages** from the menu bar.



When opening the Studio for the first time, the perspective displayed will depend on the license used, so it might not be the **Integration** perspective at first. For example, if you are using a Master Data Management license, the **MDM** perspective will open first. To display the Welcome page, you will have to switch to the **Integration** perspective.

1.1.2. How to connect to TalendForge

Every fourth time you launch *Talend Studio*, until you are connected to the **Talend** Community, the **[Connect to TalendForge]** dialog box opens, inviting you to connect to the **Talend** Community so that you can check, download, install external components and upload your own components to the **Talend** Community to share with other **Talend** users directly in the **Exchange** view of your Job designer in the Studio.

To learn more about the **Talend** Community, click the **TalendForge Terms of Use** link. For more information on using and sharing community components, see the section on how to download/upload **Talend** community components of your Studio User Guide.

If you want to connect to the **Talend** Community later, click **Skip this Step** to continue launching the Studio without setting up a connection to the **Talend** Community.

1. By default, the Studio will automatically collect product usage data and send the data periodically to servers hosted by **Talend** for product usage analysis and sharing purposes only. If you do not want the Studio to do so, clear the **I want to help to improve Talend by sharing anonymous usage statistics** check box.

You can also turn on or off usage data collection from the **[Preferences]** dialog box (**Talend > Usage Data Collector**). For more information, see the section on setting *Talend Studio* preferences of your Studio User Guide.

2. Fill in the required information, select the **I Agree to the TalendForge Terms of Use** check box, and click **CREATE ACCOUNT** to create your account and connect to the **Talend** Community automatically and continue launching the Studio.



Be assured that any personal information you may provide to **Talend** will never be transmitted to third parties nor used for any purpose other than joining and logging in to the **Talend** Community and being informed of **Talend** latest updates.

Connect to TalendForge

Connect your Studio to TalendForge, the Talend **Online Community**.

- Download **new components and connectors** from Talend Exchange.
- Access the most recent **Documentation and Tech articles** from Talend social knowledgebase.
- See the latest messages in the Talend **Discussions Forums**.

talend
FORGE

user

user@comapny.com

.....

.....

United States

☒ I agree to the [TalendForge Terms of Use](#)

☒ I want to help to improve Talend by sharing anonymous usage statistics

CREATE ACCOUNT

Connect to Existing Account Skip this Step

If you already have created an account at <http://www.talendforge.org>, click **Connect to Existing Account**, fill in your user name and password, and click **CONNECT TO MY ACCOUNT** to sign in the **Talend** Community and continue launching the Studio.

Connect your Studio to TalendForge, the Talend Online Community.

- Download **new components and connectors** from Talend Exchange.
- Access the most recent **Documentation and Tech articles** from Talend social knowledgebase.
- See the latest messages in the Talend **Discussions Forums**.



☒

☒

☒ I want to help to improve Talend by sharing anonymous usage statistics

CONNECT TO MY ACCOUNT

Create a New Account

Skip this Step



This page will not appear again when the Studio starts up once you successfully connect to the **Talend** Community. To show this page again, select **Talend > Exchange** from the **[Preferences]** dialog box, and click Sign In. For more information, see the section on setting *Talend Studio* preferences of your Studio User Guide.

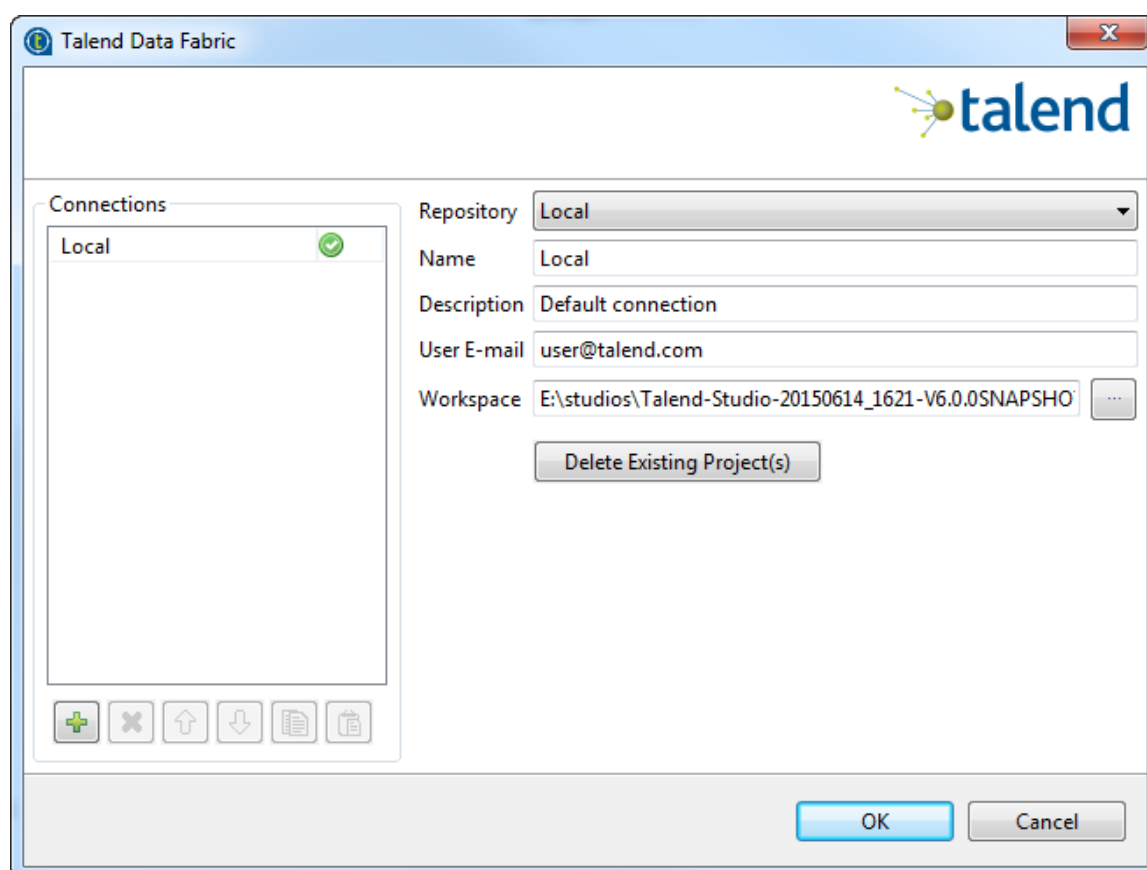
1.1.3. How to access a Repository

When launching *Talend Studio*, you can connect to a local repository where you store the data for your projects, including Jobs and business models, metadata, routines, etc. You can also connect to a remote repository where you store the same type of data to work collaboratively on projects.

1.1.3.1. How to connect to a local repository

To set a connection to a local repository, do the following:

1. On the login window of *Talend Studio*, click the **Manage Connections** button to open the repository connection setup dialog box.



Depending on the license you are using, the product information displayed in your Studio may differ slightly from what is shown above.

2. From the **Repository** list, select **Local**.
3. If needed, type in a name and a description for your connection in the relevant fields.
4. In the **User E-mail** field, type in the email address that will be used as your user login. This field is compulsory to be able to use *Talend Studio*.

Be aware that the email entered is never used for purposes other than logging in.

5. By default, the **Workspace** field shows the path to the current workspace directory which contains all of the folders belonging to the project created. To change the workspace directory, type in the name of an existing directory or click the [...] button next to the **Workspace** field and browse to your preferred workspace directory. Upon changing your workspace directory, unless it is the first startup, you need to restart your *Talend Studio* by clicking the **Restart** button back on the login window for your change to take effect.

For more information about workspace directories, see [Working with different workspace directories](#).

6. If needed, click the plus [+] button in the lower left corner and set the connection information to add as many connections as needed.

To edit a connection, select the connection and edit the connection details following the steps above.

To remove a connection, select the connection and click the [X] button.

7. Click **OK** to validate your changes and return to the login window.

1.1.3.2. How to connect to a remote repository



Before connecting to the remote repository for the first time, make sure you received a login and password from the administrator who created your user account stored in *Talend Administration Center*.

Note that, for users who loaded a license from a remote server when they first launched the Studio, the information related to the connection to a remote repository are retrieved from *Talend Administration Center* and thus are automatically completed.

To connect manually to a remote repository, do the following:

1. On the login window of *Talend Studio*, click the **Manage Connections** button to open the repository connection setup dialog box.

To access a remote repository, configure the connection access in the following steps.

2. Add a connection to the **Connections** list using the [+] button.
3. From the **Repository** list, select **Remote**.

In the **Name** field, enter a name to the connection. This name will be displayed in the connection list on the login window of *Talend Studio*

If needed, enter a description of your connection in the **Description** field.

In the **User E-mail** and **User Password** fields, enter the user details you received from the administrator.

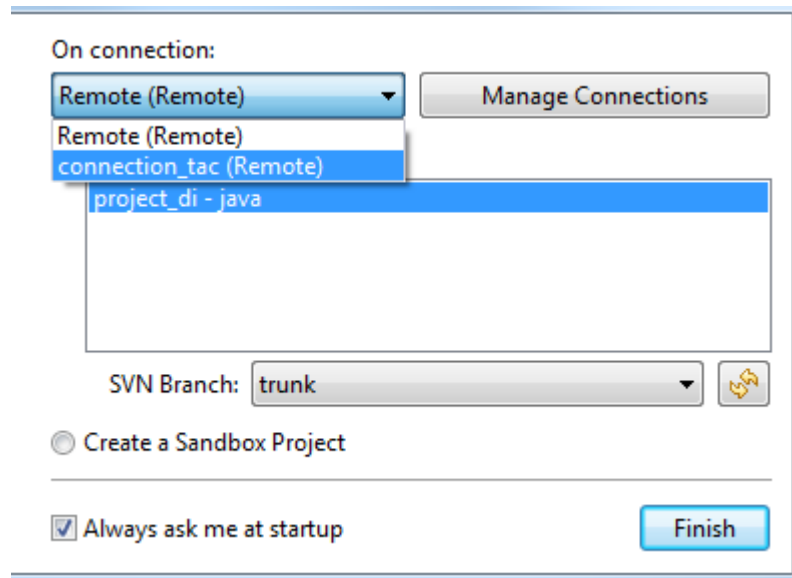
By default, the **Workspace** field shows the path to the current workspace directory. To change the workspace directory, type in the name of an existing directory or click the [...] button next to the **Workspace** field and browse to your preferred workspace directory. Upon changing your workspace directory, unless it is the first startup, you need to restart your *Talend Studio* by clicking the **Restart** button back on the login window for your change to take effect.

For more information about workspace directories, see [Working with different workspace directories](#).

In the **Web-app URL** field, enter the address of the web application where the repository is stored, as provided by your **Talend** administrator, and click the **Check url** button to verify your login connection.

4. Click **OK** to validate your connection settings and go back to login window.

On the login window, the newly created connection is displayed on the **Connection** list.



As soon as you are connected with *Talend Administration Center* and if an update for your Studio is found, an **update** button appears at the bottom of the login window and the **Open** button becomes inoperable. Click **update** to download and install the update. When the installation completes, click the **restart** button that appears next to the **update** button to restart your Studio so that the newly installed update takes effect. For more information on the software update process, see the *Talend Installation Guide*.

5. From the **Connection** list, select the relevant connection.
6. Click the **Refresh** button to update the existing project list.
7. From the **Project** list, select the project you want to launch.
8. From the **Branch** list, select the trunk, a branch, or a tag, whichever is desired.



A tag is a read-only copy of an SVN or Git managed project. If you choose to open a tag, you can make changes to your project items but you will be unable to permanently save your changes to a Job unless you copy the Job to a branch or the trunk. For how to copy a Job to a branch, see your Studio User Guide.

9. Click **Finish** to launch the project selected in *Talend Studio*. The listed projects are the projects allocated to you in *Talend Administration Center*.



According to your user **Role** defined in *Talend Administration Center*, the **Create**, **Import** and **Demos** features are enabled or disabled.

1.1.4. How to set up a project in the repository

To open *Talend Studio*, you must first set up a project in the repository you connected to earlier.

You can set up a project in the repository by:

- creating a new project. For more information, see [How to create a project](#).

- importing one or more local projects you already created in other sessions of *Talend Studio*. For more information, see *Talend Studio User Guide*.
- importing the Demo project. For more information, see [How to import a demo project](#).

1.2. Working with different workspace directories

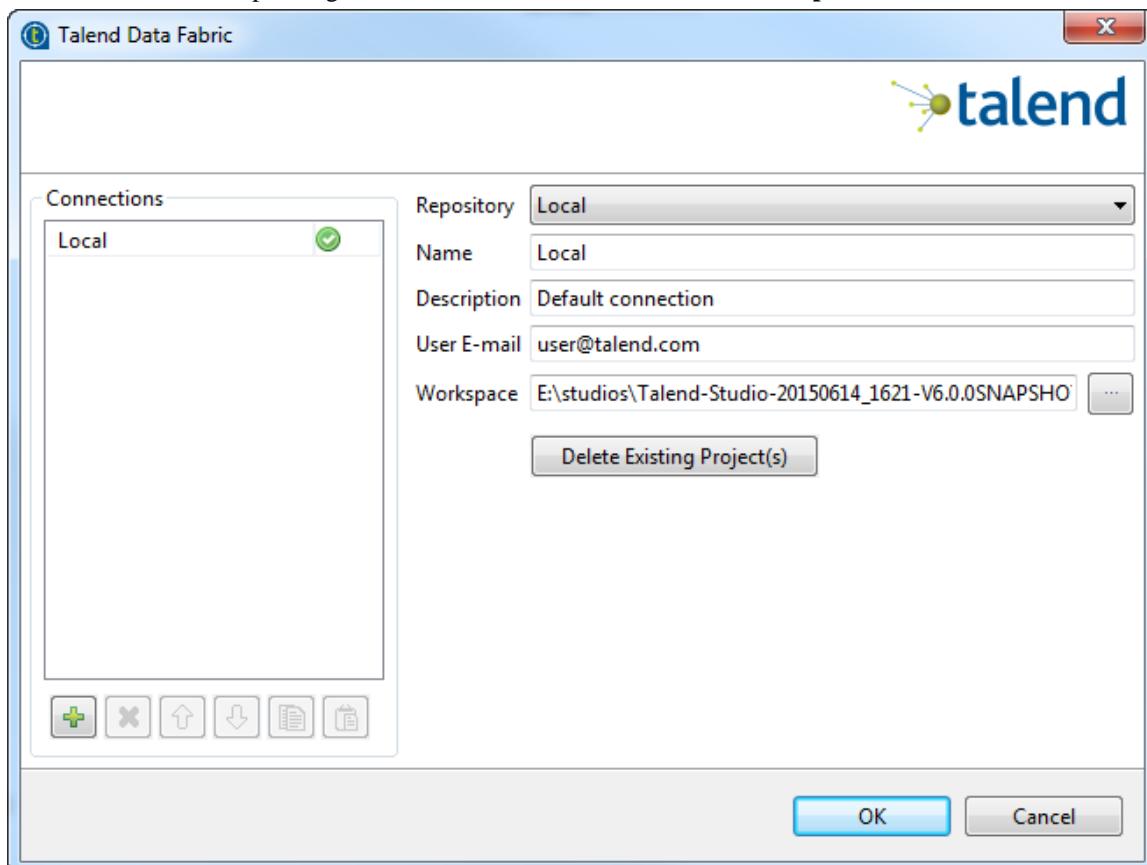
Talend Studio makes it possible to create many workspace directories and connect to a workspace different from the one you are currently working on, if necessary.

This flexibility enables you to store these directories wherever you want and give the same project name to two or more different projects as long as you store the projects in different directories.

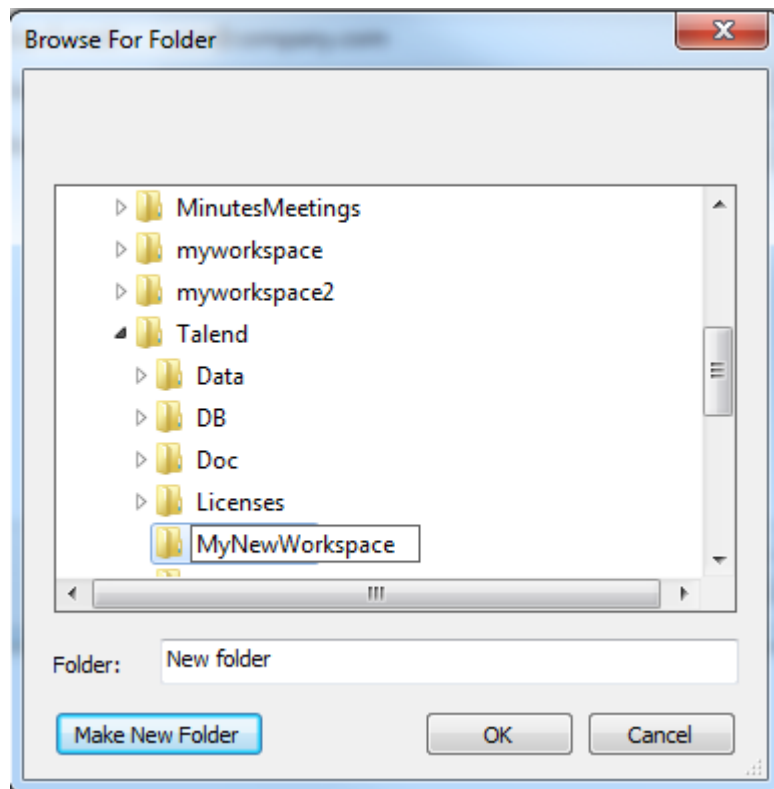
1.2.1. How to create a new workspace directory

Talend Studio is delivered with a default workspace directory. However, you can create as many new directories as you want and store your project folders in them according to your preferences.

1. If you have already started the Studio, select **File > Switch Project or Workspace** from the menu bar to restart the Studio.
2. On the login window, click **Manage Connections** to open the connection setup dialog box.
3. On the connection setup dialog box, click the [...] button next to the **Workspace** field.



4. In the **[Browse For Folder]** dialog box, browse to the parent directory under which you want to create a new workspace directory, click **Make New Folder**, and enter the name of your new workspace directory. Then click **OK** to validate directory creation and close the dialog box.

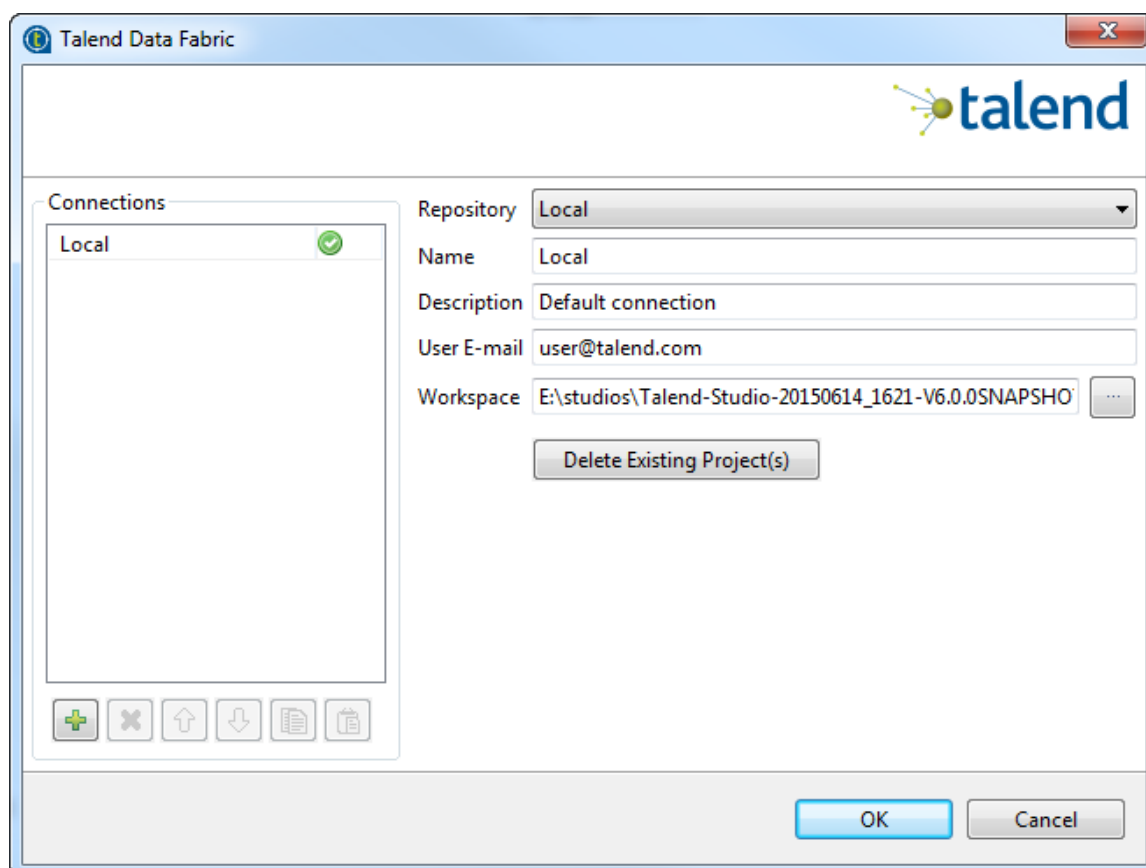


5. Click **OK** to validate your connection setup and go back to the login window.
6. Back on the login window, click the **Restart** button to restart your *Talend Studio* for the change to take effect.

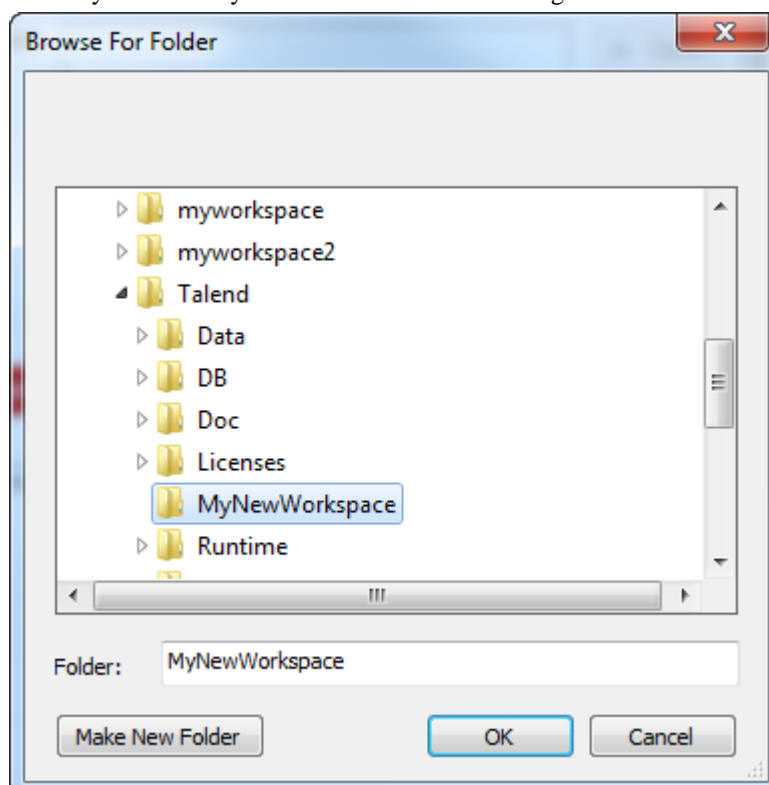
1.2.2. How to connect to a different workspace directory

In *Talend Studio*, you can select the workspace directory you want to store your project folders in according to your preferences.

1. If you have already started the Studio, select **File > Switch Project or Workspace** from the menu bar to restart the Studio.
2. On the login window, click the **Manage Connections** button to open the connection setup dialog box.
3. On the connection setup dialog box, click the [...] button next to the **Workspace** field.



4. In the **[Browse For Folder]** dialog box, browse to your preferred folder to use as the new workspace directory, and click **OK** to validate your directory selection and close the dialog box.



5. Click **OK** to validate your connection setup and go back to the login window.

6. Back on the login window, click the **Restart** button to restart your *Talend Studio* for the change to take effect.

1.3. Working with projects

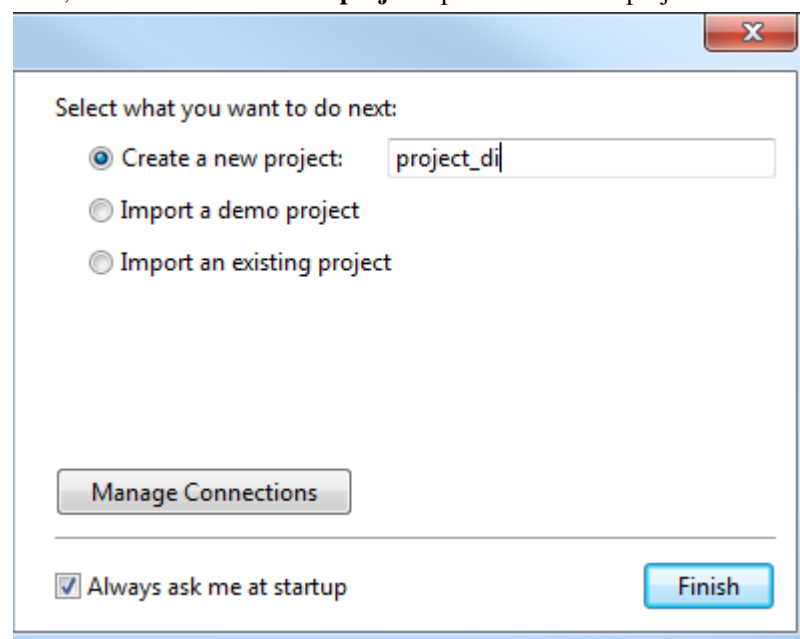
In *Talend Studio*, the highest physical structure for storing all different types of data integration Jobs, metadata, routines, and so on is the "project".

This section will guide you through the basic steps to manage projects before starting your work on Business Models, Jobs, Routes, and so on in your *Talend Studio*. For more information on project management, see your *Talend Studio User Guide*.

1.3.1. How to create a project

To create a local project at the initial startup of the Studio, do the following:

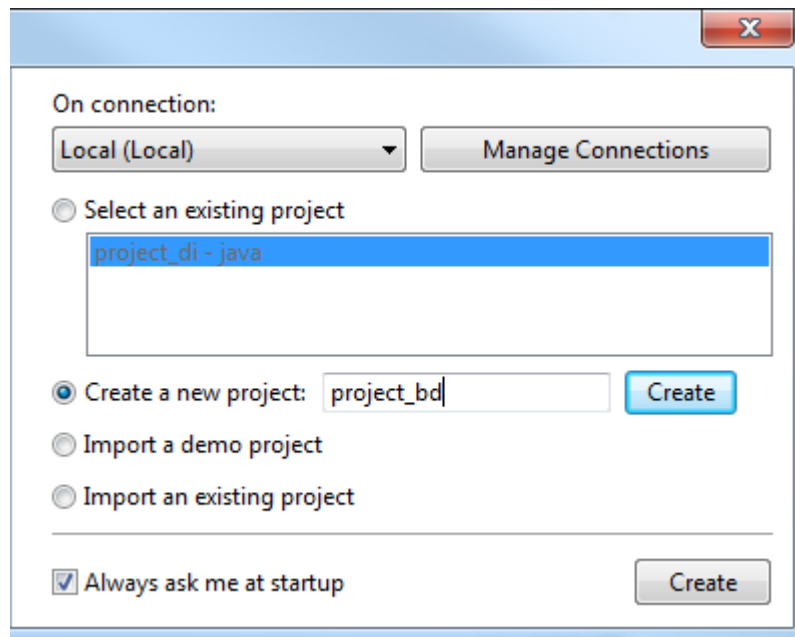
1. Launch *Talend Studio* and connect to a local repository.
2. On the login window, select the **Create a new project** option and enter a project name in the field.



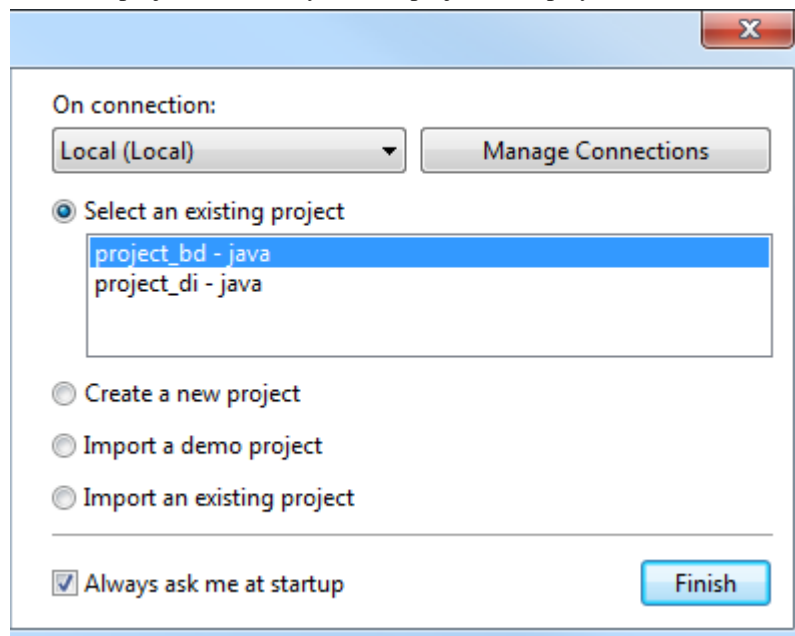
3. Click **Finish** to create the project and open it in the Studio.

To create a new local project after the initial startup of the Studio, do the following:

1. On the login window, select the **Create a new project** option and enter a project name in the field.



- Click **Create** to create the project. The newly created project is displayed on the list of existing projects.



- Select the project on the list and click **Finish** to open the project in the Studio.

Later, if you want to switch between projects, on the Studio menu bar, use the combination **File > Switch Project or Workspace**.

1.3.2. How to create a sandbox project

A sandbox project is a working project created from *Talend Studio* by a new user not registered in *Talend Administration Center* to test data, Jobs, environments, etc. When you as a new user create a sandbox project, you create both your project in a remote repository and your user account on *Talend Administration Center*. This way, the project can be easily shared with other users and migrated to a production environment.



If your account already exists in Talend Administration Center, you will not be able to create a sandbox project.

To create a sandbox project:

1. Launch *Talend Studio* using a remote connection.
2. On login screen, select **Create a Sandbox Project** and click **Select**. A **[Create Sandbox project]** dialog box opens.

3. In the **URL** field, type in the URL of *Talend Administration Center*.
To get *Talend Administration Center*'s URL, contact your system administrator.
4. Click **Check** to validate *Talend Administration Center*'s URL.
5. In the **Login** and **Password** fields, type in the email address and password that will be used to connect to your remote project with *Talend Studio* and to connect to *Talend Administration Center* if you want to change your password, for example.

Be aware that the email entered is never used for another purpose other than logging in.



If your account already exists in Talend Administration Center, you will not be able to create a sandbox project.

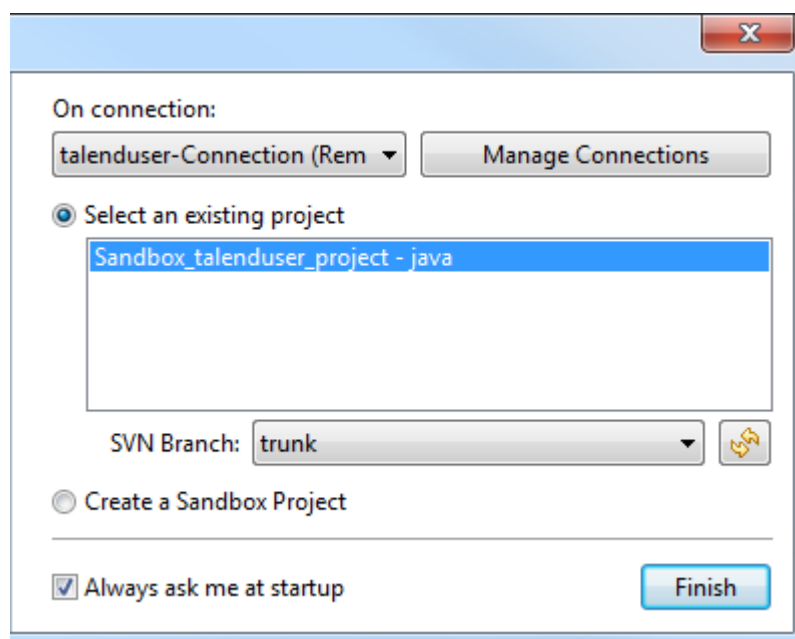
6. In the **First name** and **Last name** fields, type in your first and last name.
7. Click **OK** to validate.

A popup window prompts you to indicate that your sandbox project and its corresponding connection have successfully been created. There are respectively named **Sandbox_username_project** and **username_Connection**

8. Click **OK** to close the popup.

You might receive an email notifying of your account creation on *Talend Administration Center*, if the administrator activated this functionality.

The **Connection**, **Email** and **Password** fields are automatically filled in with the connection information you provided and the **Project** list is automatically filled in with your newly created sandbox project.




To open the newly created sandbox project in *Talend Studio*, select your Sandbox project connection from the connection list, select the project list, and click **Finish**.

1.3.3. How to import a demo project

You can import one or more demo projects that include numerous samples of ready to use Jobs into your *Talend Studio* to help you understand the functionalities of different **Talend** components.

To import a demo project, proceed as follows:

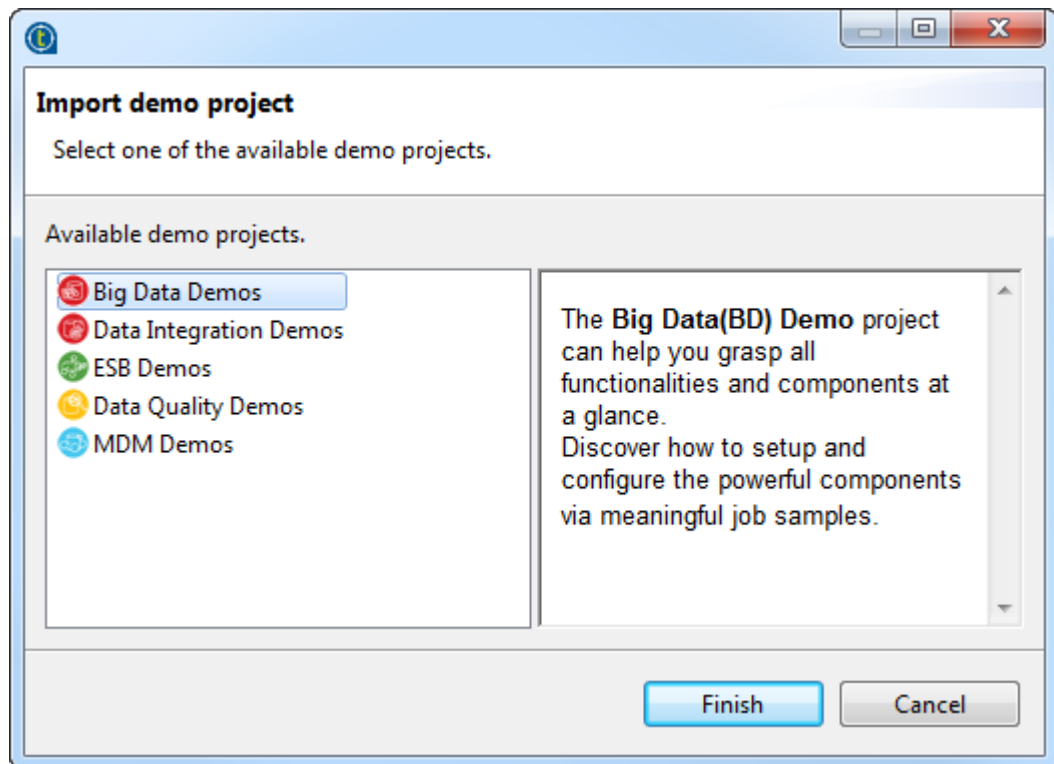
1. When launching your *Talend Studio*, select the **Import a demo project** option on the Studio login window and click **Select**, or click the **Demos** link on the welcome window, to open the **[Import demo project]** dialog box.

After launching the Studio, click  button on the toolbar, or select **Help > Welcome** from the Studio menu bar to open the welcome window and then click the **Demos** link, to open the **[Import demo project]** dialog box.

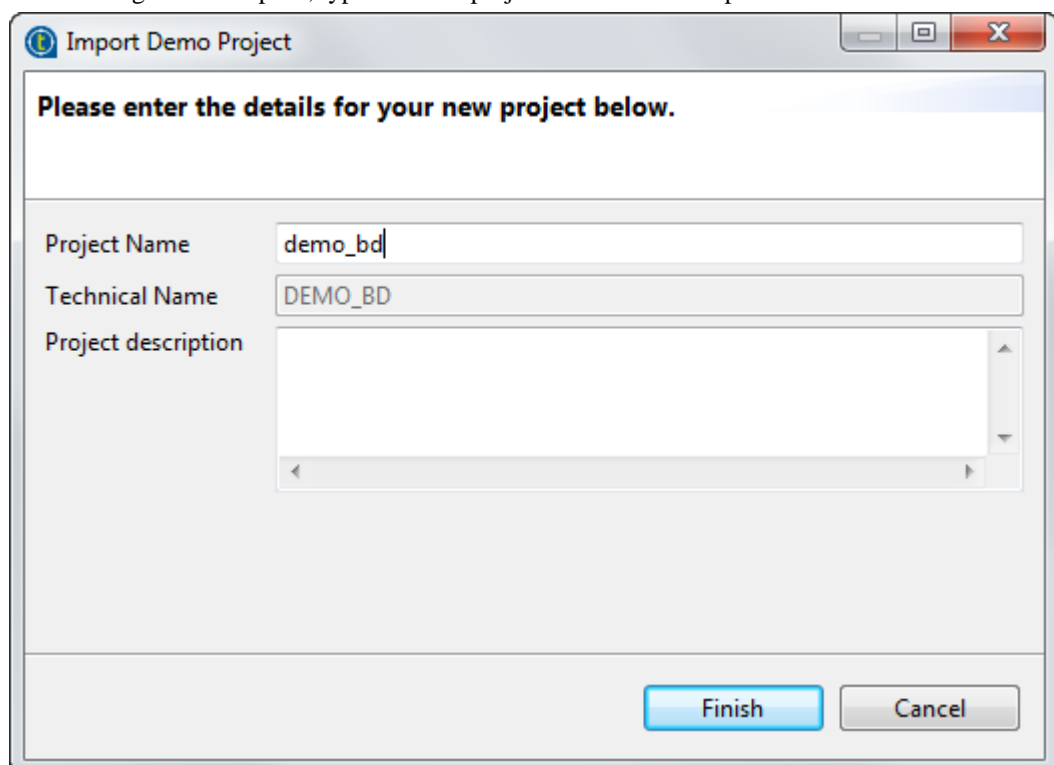
2. In the **[Import Demo Project]** dialog box, select the demo project you want to import and view the description on the right panel.



The demo projects available in the dialog box may vary depending on the license you are using.



3. Click **Finish** to close the dialog box.
4. In the new dialog box that opens, type in a new project name and description information if needed.



5. Click **Finish** to create the project.

All the samples of the demo project are imported into the newly created project, and the name of the new project is displayed in the **Project** list on the login screen.

- To open the imported demo project in *Talend Studio*, back on the login window, select it from the **Project** list and then click **Finish**.

The Job samples in the open demo project are automatically imported into your workspace directory and made available in the **Repository** tree view under the **Job Designs** folder.

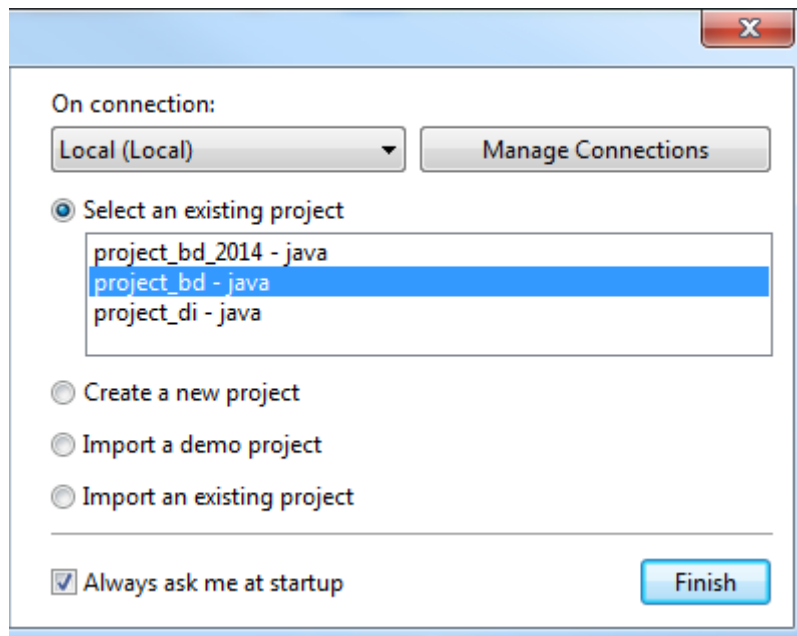
1.3.4. How to open a local project



When you launch Talend Studio for the first time, no project names are displayed on the **Project** list. First you need to create a project or import a local or Demo project in order to populate the **Project** list with the corresponding project names that you can then open in the Studio.

To open a local project in *Talend Studio*:

On the Studio login screen, select the connection to the local repository that holds your project from the connection list, select the project of interest from the project list and click **Finish**.



A progress bar appears. Wait until the task is complete and the *Talend Studio* main window opens.



When you open a project imported from a previous version of the Studio, an information window pops up to list a short description of the successful migration tasks.

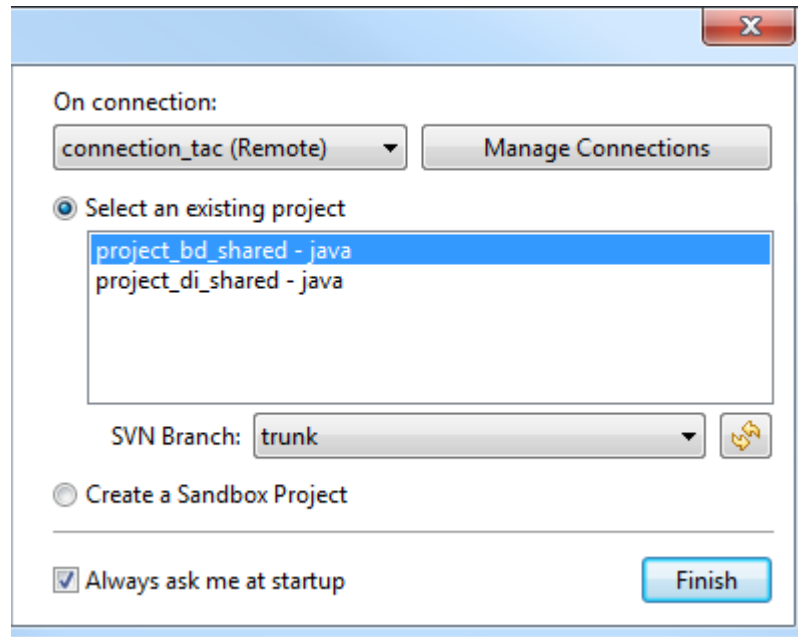
1.3.5. How to open a remote project



To open a remote project, you must first create a connection to the repository on which the project is stored and make sure you have access rights to the project. For further information on creating a connection to a remote repository, see [How to connect to a remote repository](#).

To open a remote project in *Talend Studio*:

- On the **Connection** area of the Studio login window, select the connection to the repository in which the project is stored from the **Connection** list.



As soon as you are connected with *Talend Administration Center* and if an update for your Studio is found, an **update** button appears at the bottom of the login window and the **Open** button becomes inoperable. Click **update** to download and install the update. When the installation completes, click the **restart** button that appears next to the **update** button to restart your Studio so that the newly installed update takes effect. For more information on the software update process, see the *Talend Installation Guide*.

2. Click the **Refresh** button to update the list of existing projects, which are the projects allocated to you in *Talend Administration Center*.

Note that, if an administrator edits your access rights on a project while you are already connected to this project in the Studio, you have to relaunch the Studio to take these rights into account.

3. From the project list, select the project you want to open.
4. From the **Branch** list, select the trunk (SVN only) or master (Git only), a branch, or a tag, whichever is desired.




A tag is a read-only copy of an SVN or Git managed project. If you choose to open a tag, you can make changes to your project items but you will be unable to permanently save your changes to a Job unless you copy the Job to a branch or the trunk. For how to copy a Job to a branch, see your *Studio User Guide*.

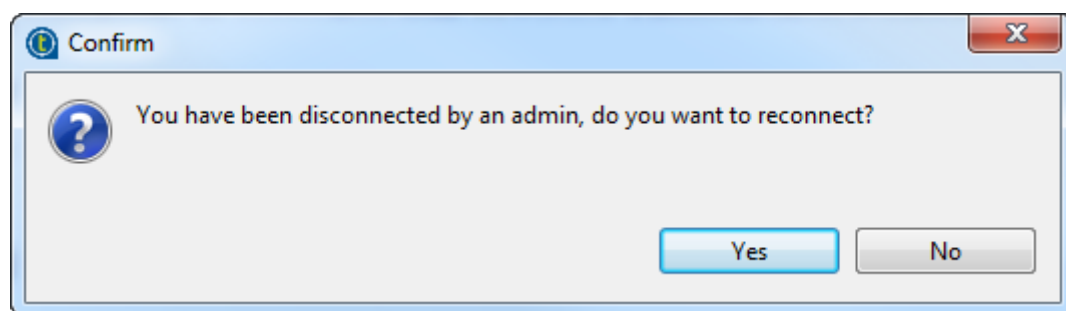
5. Click **Finish** to launch the selected project in the Studio.

A progress bar appears, and the *Talend Studio* main window opens. A generation engine initialization dialog box displays. Wait until the initialization is complete.

Upon opening a remote project, *Talend Studio* checks periodically its connection with *Talend Administration Center*.

When *Talend Studio* detects loss of connection, it tries automatically to reconnect to *Talend Administration Center*. You can view the connection progress on the **Progress** tab by double-clicking **Check Administrator connection** at the lower right corner of the *Talend Studio* main window. If you click the  button at this phase, the project will enter the read-only mode.

Once *Talend Studio* detects that you have been logged out by an administrator in *Talend Administration Center*, a confirmation dialog box appears asking you whether to reconnect to *Talend Administration Center*.



Click **Yes** to reconnect to *Talend Administration Center*. *Talend Studio* will perform an authorization check when trying a reconnection. A warning will be displayed and the project will enter the read-only mode if:

- you no longer have access to the project you have opened, or
- you no longer have access to any reference project of the project you have opened, or
- the number of reference projects of the project you have opened has changed.

If your access right to the project you have opened has changed from read-write to read-only, or if you click **No** in the confirmation dialog box, the project directly goes into the read-only mode.

When the project is in the read-only mode, you can still edit the Job or Jobs currently open in the design workspace, and changes you make will be committed to the SVN or Git the next time you log in to *Talend Administration Center* with read-write access to the project.

1.4. Managing licenses

When you subscribe to *Talend Studio*, you receive a license that will authorize you to use the Studio.

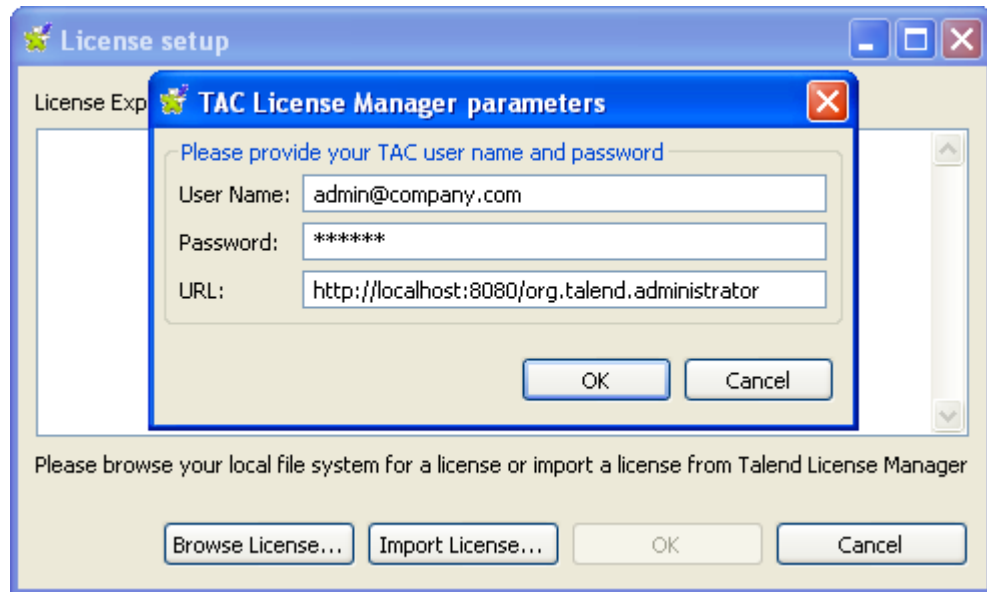
1.4.1. Setting a license for the Studio

When you launch the Studio for the first time, or if the license for the studio you try to launch has expired, a dialog box appears prompting you to set a license.

To set a license for the studio, do the following:

1. In the **[License setup]** dialog box, click **Browse License...** to browse and select your license file.

If you have already set your license and project in *Talend Administration Center* web application, click **Import License...** to retrieve the license. To do this, fill in the credentials and the URL of your *Talend Administration Center* web application.



This way, you do not have to set up a remote repository as the settings of the project you created in the Web application are automatically retrieved.

A sentence is displayed at the bottom of the dialog box to confirm that the license you have loaded is valid.



The studio license comes with your subscription for a specific product. It determines what perspectives you can have access to. For more information about perspectives, see [Multi-perspective approach](#).

2. Click **OK** to close the dialog box and continue launching the studio or **Cancel** to cancel the operation.

A license file is automatically created in the root directory of the studio.

1.4.2. Checking and replacing the license for the Studio

To replace the license you are using in the studio with a new license, proceed as follows:

1. From the studio menu bar, select **Help > About License**.

The **[About License]** dialog box is displayed.

Please import your product license from Administration Center or browse your local file system for it:

☒ My product license is on a remote host:

Login:

Password:

Server URL:

☐ My product license is on the local file system:

2. Either:

- Click **Fetch** to import a license from *Talend Administration Center*.

Make sure to define your connection information to *Talend Administration Center* correctly before you import the license.

- Or, select the **My product license is on the local file system** and browse to the license.

3. Click **Next**.

A confirmation message is displayed prompting you to restart the studio with the new license.

4. Click **OK** to close the message and restart the studio.

The license file is automatically updated in the root directory of the studio.

1.5. Multi-perspective approach

Talend Studio offers a comprehensive set of tools and functions for all its key capabilities including data and application integration, data profiling and master data management. These tools are all accessible from different perspectives within the studio.



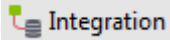
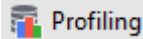
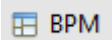
The availability of perspectives in the studio depends on the license you have when you are working in a local project, or on the type of the remote project itself when you are working in remote projects. For further information about licenses, see [Managing licenses](#).

1.5.1. Switching between different perspectives

There are different ways to switch between different perspectives in the studio. They are as follows:

To switch between perspectives using quick access icons, do the following:

- In the top right corner of the studio, select:

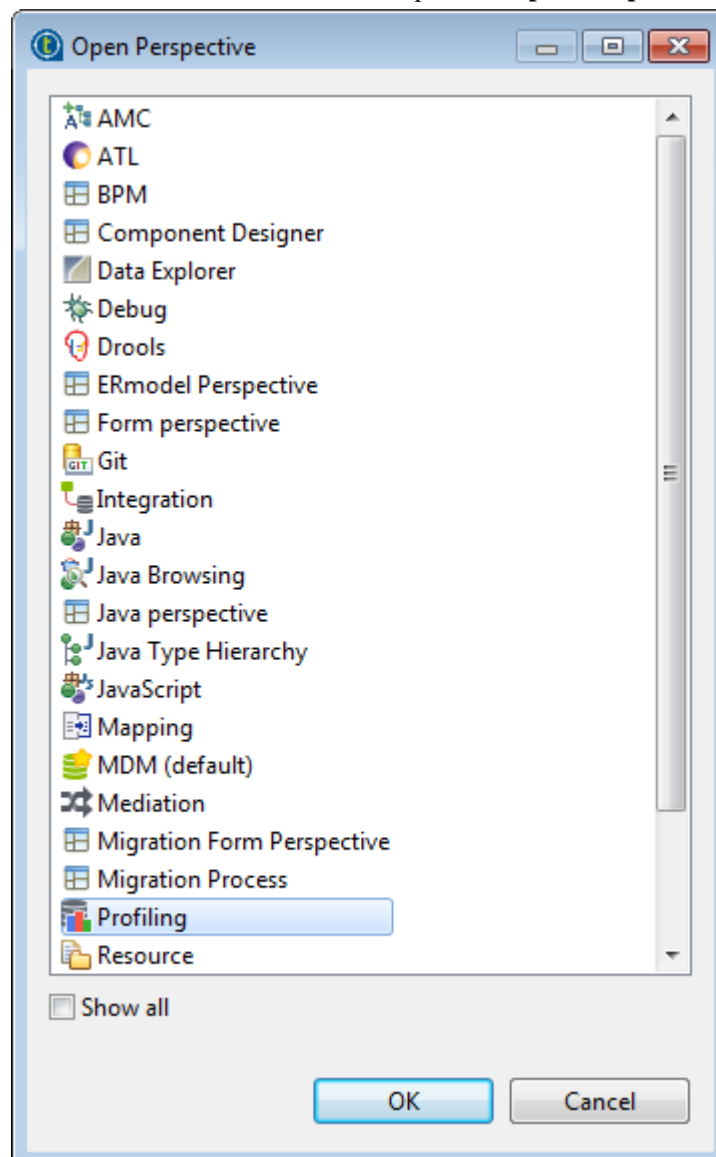
Icon	to...
 Integration	open the Integration perspective where you have access to a set of components and routines dedicated to data integration.
 Profiling	open the Profiling perspective where you can examine data in different data sources and design data cleansing analyses.
 BPM	open the BPM perspective where you can design business workflows using graphical tools.



The availability of the above perspectives in the studio depends on the license you have when you are working in a local project, or on the type of the remote project itself when you are working in remote projects.

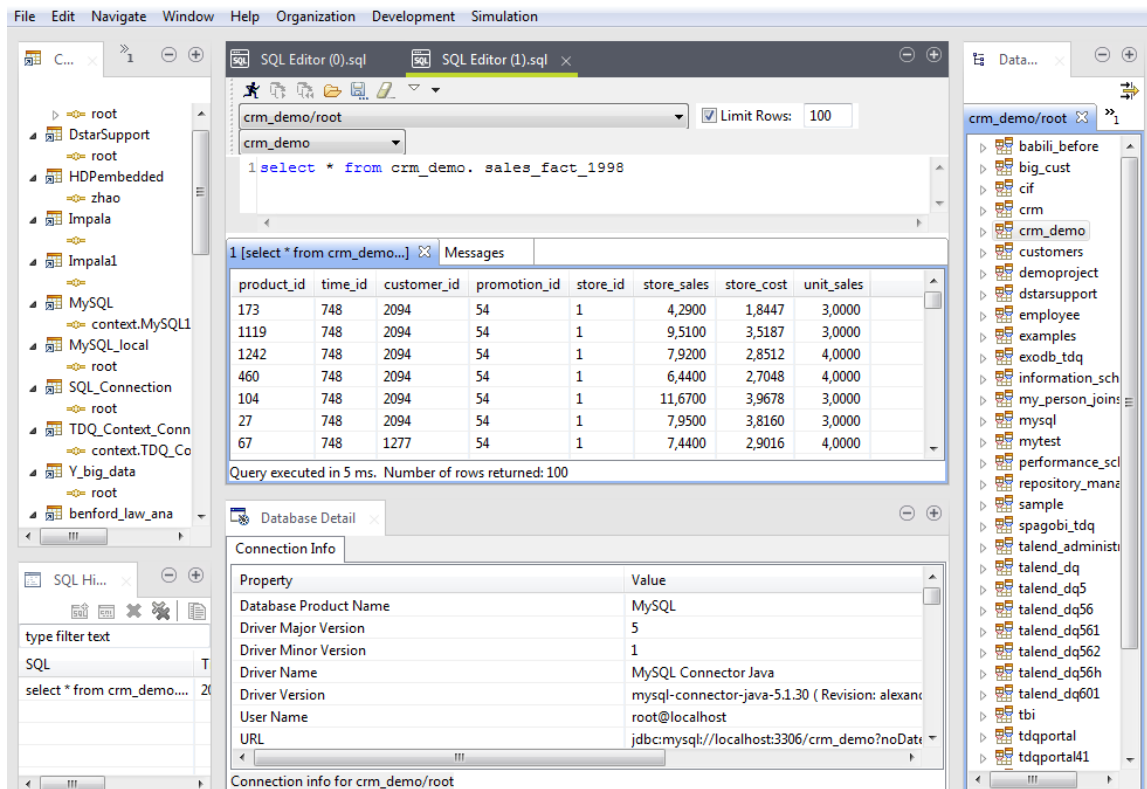
It is also possible to use the **[Open Perspective]** dialog box as the following:

1. In the top right corner of the studio, click the  icon to open the **[Open Perspective]** dialog box.



2. Select the perspective you want to access and then click **OK**.

The selected perspective opens in the studio and an icon is docked in the top right corner.

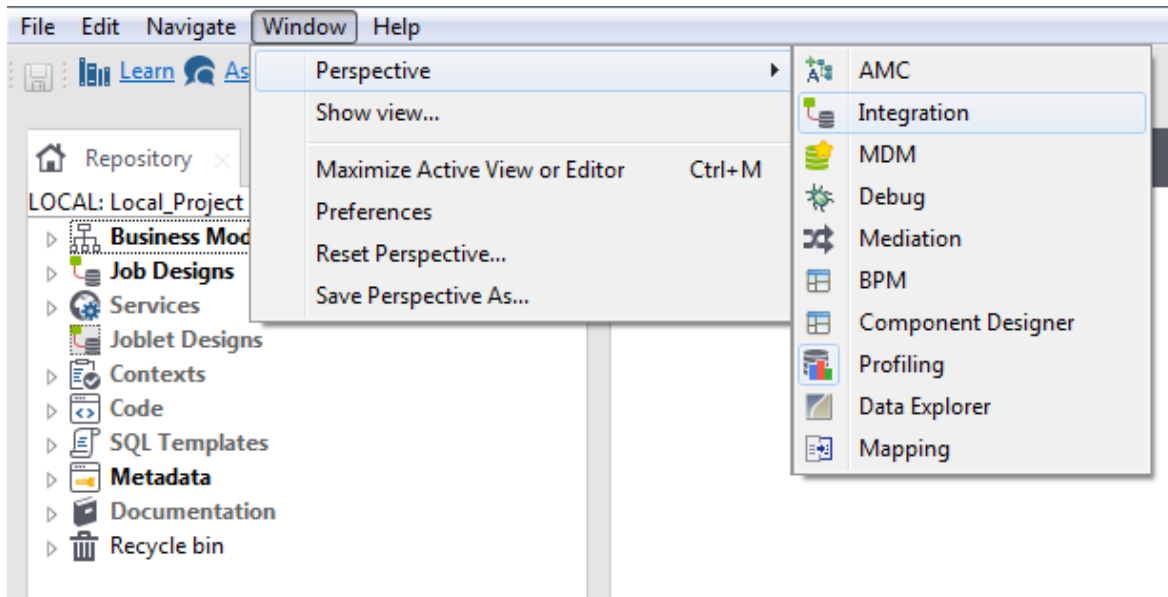


- Do the same to open other perspectives from the dialog box and add quick access icons for them.

You can manage the display of the quick access icons. For further information, see [Managing quick access icons](#).

Alternatively, you may switch between perspectives using the menu bar:

- On the menu bar, click **Window > Perspective**.



- Select from the list the perspective you want to open in the studio.

An icon for the perspective is docked in the top right corner of the studio.

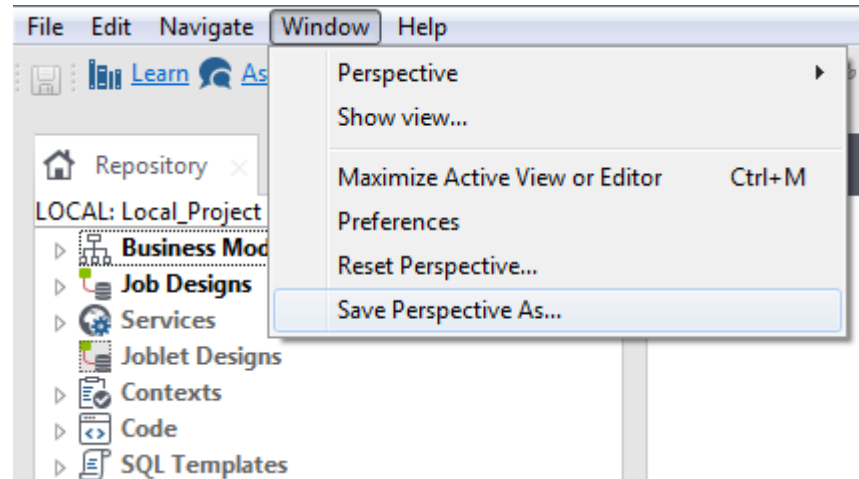
It is also possible, using the **Window - Show view...** combination, to show views from other perspectives in the open perspective.

1.5.2. Saving the configuration of a perspective

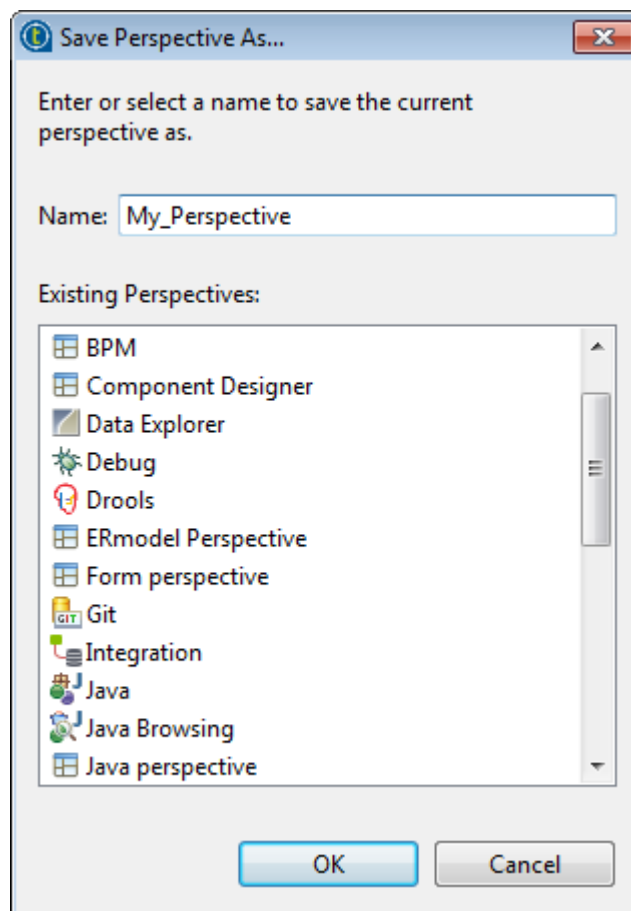
You can save the configuration of your current perspective in order to list it as a new perspective in the perspective dialog box.

To save the configuration of the current perspective, do the following:

1. On the menu bar, click **Window > Save Perspective As....**



2. In the **Name** field, enter a name.



3. Click **OK**.

The current perspective is saved as a new perspective under the new name.

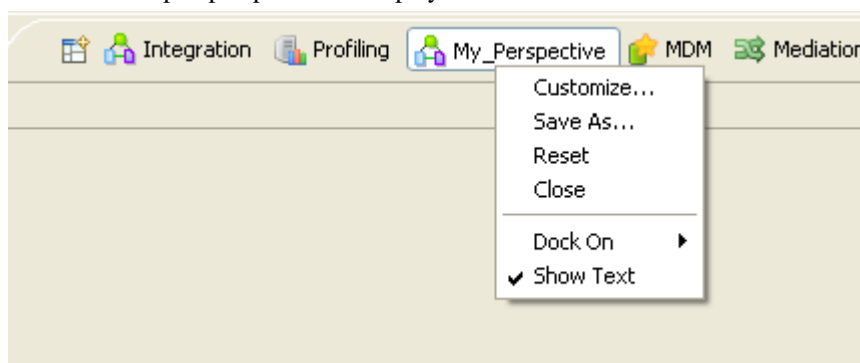
You can open this perspective any time by selecting it from the **[Open Perspective]** dialog box. For further information, see [Switching between different perspectives](#).

1.5.3. Managing quick access icons

You can manage the icons in the top right corner of the studio and adapt them to your personal use.

To manage the quick access icons, do the following:

1. Right-click the icon of the open perspective to display a contextual menu.



2. Select an option from the list as the following:

Option	Description
Save As...	Saves the current perspective under a different name.
Reset	Puts the current perspective back to its default state.
Close	Closes the current perspective.
Show text	Displays or hides the text next to the icon



Chapter 2. Working in *Talend Studio* - basic data integration Job examples

This chapter provides basic data integration Job examples to help users get started with *Talend Studio*. For more real-life examples, see the *Theory into practice* chapter of your *Talend Studio User Guide*.

2.1. Getting started with a basic Job

This section provides a continuous example that will help you create, add components to, configure, and execute a simple Job. This Job will be named *A_Basic_Job* and will read a text file, display its content on the **Run** console, and then write the data into another text file.

2.1.1. Creating a Job

To create the example Job described in this section, proceed as follows:

1. In the **Repository** tree view of the **Integration** perspective, right-click the **Job Designs** node and select **Create Job** from the contextual menu.

The **[New Job]** wizard opens to help you define the main properties of the new Job.

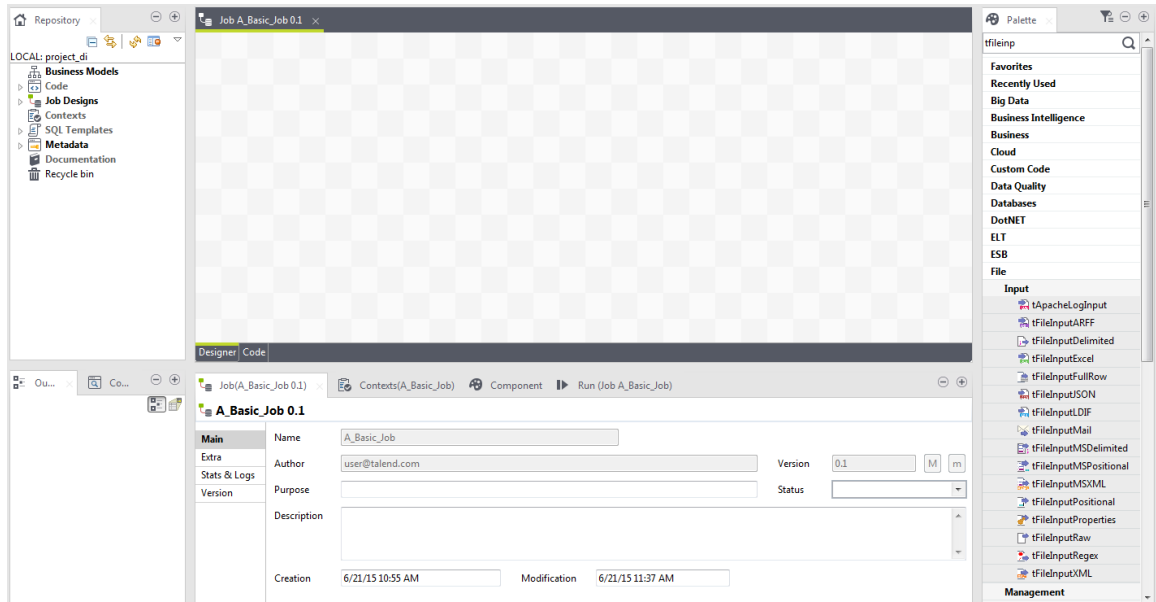
2. Fill the Job properties as shown in the previous screenshot.

The fields correspond to the following properties:

Field	Description
Name	the name of the new Job. Note that a message comes up if you enter prohibited characters.
Purpose	Job purpose or any useful information regarding the Job use.
Description	Job description containing any information that helps you describe what the Job does and how it does it.
Author	a read-only field that shows by default the current user login.

Field	Description
Locker	a read-only field that shows by default the login of the user who owns the lock on the current Job. This field is empty when you are creating a Job and has data only when you are editing the properties of an existing Job.
Version	a read-only field. You can manually increment the version using the M and m buttons.
Status	a list to select from the status of the Job you are creating.
Path	a list to select from the folder in which the Job will be created.

3. An empty design workspace opens up showing the name of the Job as a tab label.



The Job you created is now listed under the **Job Designs** node in the **Repository** tree view.

You can open one or more of the created Jobs by simply double-clicking the Job label in the **Repository** tree view.

Related topics:

- Classify the Jobs you created by creating folders. For more information, see your *Talend Studio* User Guide.
- Create a data integration Job. For more information, see your *Talend Studio* User Guide.
- Create a data service Job. For more information, see your *Talend Studio* User Guide.
- Customize the workspace. For more information, see your *Talend Studio* User Guide.

2.1.2. Adding components to the Job

Now that the Job is created, components have to be added to the design workspace, a **tFileInputDelimited**, a **tLogRow**, and a **tFileOutputDelimited** in this example.

There are several ways to add a component onto the design workspace. You can:

- find your component on the **Palette** by typing the search keyword(s) in the search field of the **Palette** and drop it onto the design workspace.
- add a component by directly typing your search keyword(s) on the design workspace.
- add an output component by dragging from an input component already existing on the design workspace.
- drag and drop a centralized metadata item from the **Metadata** node onto the design workspace, and then select the component of interest from the **Components** dialog box.

This section describes the first three methods. For details about how to drop a component from the **Metadata** node, see your *Talend Studio User Guide*.

2.1.2.1. Dropping the first component from the Palette

The first component of this example will be added from the **Palette**. This component defines the first task executed by the Job. In this example, as you first want to read a text file, you will use the **tFileInputDelimited** component.

For more information regarding components and their functions, see *Talend Components Reference Guide*.

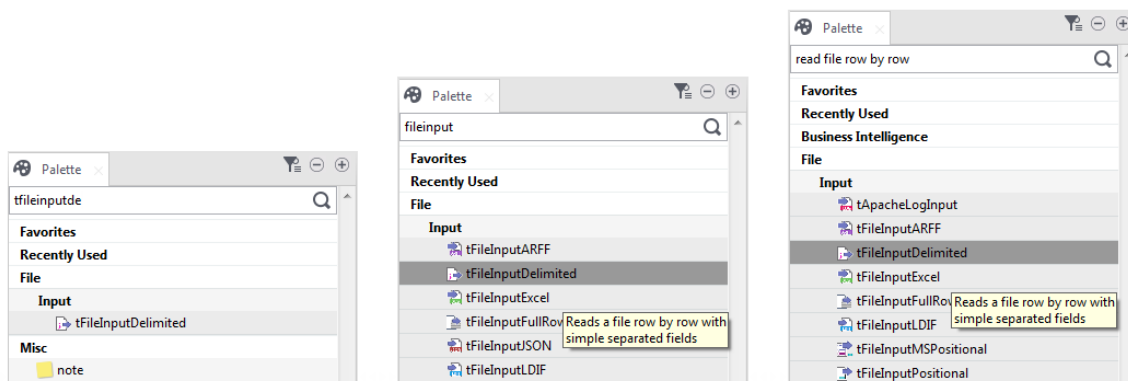
To drop a component from the **Palette**, proceed as follows:

1. Enter the search keyword(s) in the search field of the **Palette** and press **Enter** to validate your search.

The keyword(s) can be the partial or full name of the component, or a phrase describing its functionality if you don't know its name, for example, *tfileinputde*, *fileinput*, or *read file row by row*.

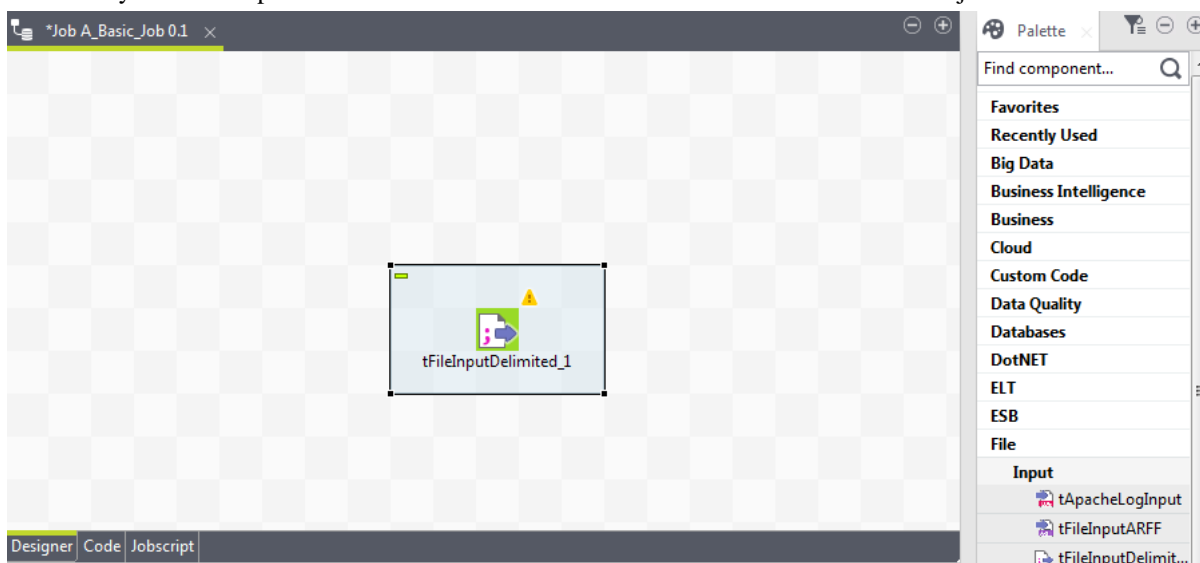


To use a descriptive phrase as keywords for a fuzzy search, make sure the **Also search from Help when performing a component searching** check box is selected on the **Preferences > Palette Settings** view. For more information, see your *Talend Studio User Guide*.



2. Select the component you want to use and click on the design workspace where you want to drop the component.

Each newly-added component is shown in a blue box to show that it as an individual Subjob.



2.1.2.2. Adding the second component by typing on the design workspace

The second component of our Job will be added by typing its name directly on the workspace, instead of dropping it from the **Palette** or from the **Metadata** node.

Prerequisite: Make sure you have selected the **Enable Component Creation Assistant** check box in the Studio preferences. For more information, see your *Talend Studio User Guide*.

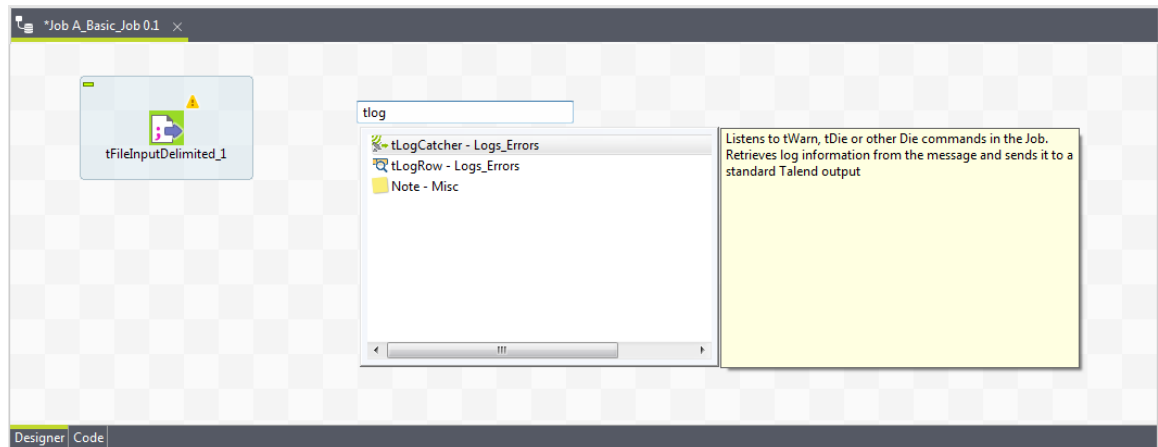
To add a component directly on the workspace, proceed as follows:

1. Click where you want to add the component on the design workspace, and type your keywords, which can be the full or partial name of the component, or a phrase describing its functionality if you don't know its name. In our example, start typing *tlog*.



To use a descriptive phrase as keywords for a fuzzy search, make sure the **Also search from Help when performing a component searching** check box is selected on the **Preferences > Palette Settings** view. For more information, see your *Talend Studio User Guide*.

A list box appears below the text field displaying all the matching components in alphabetical order.



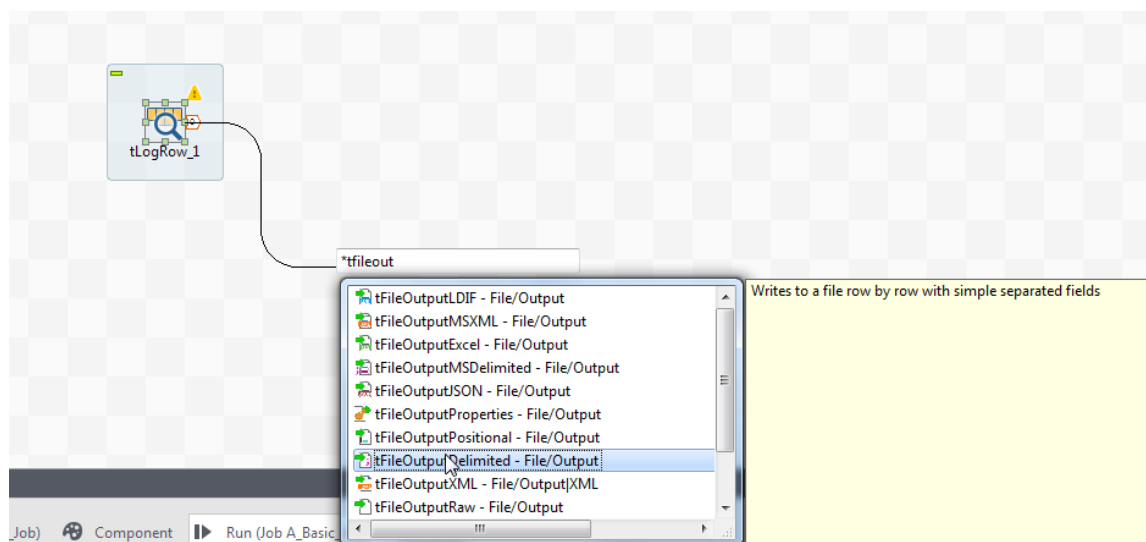
2. Double-click the desired component to add it on the workspace, **tLogRow** in our example.

2.1.2.3. Adding an output component by dragging from an input one

Now you will add the third component, a **tFileOutputDelimited**, to write the data read from the source file into another text file. We will add the component by dragging from the **tLogRow** component, which serves as an input component to the new one to be added.

1. Click the **tLogRow** component to show the **o** icon docked to it.
2. Drag and drop the **o** icon where you want to add a new component.

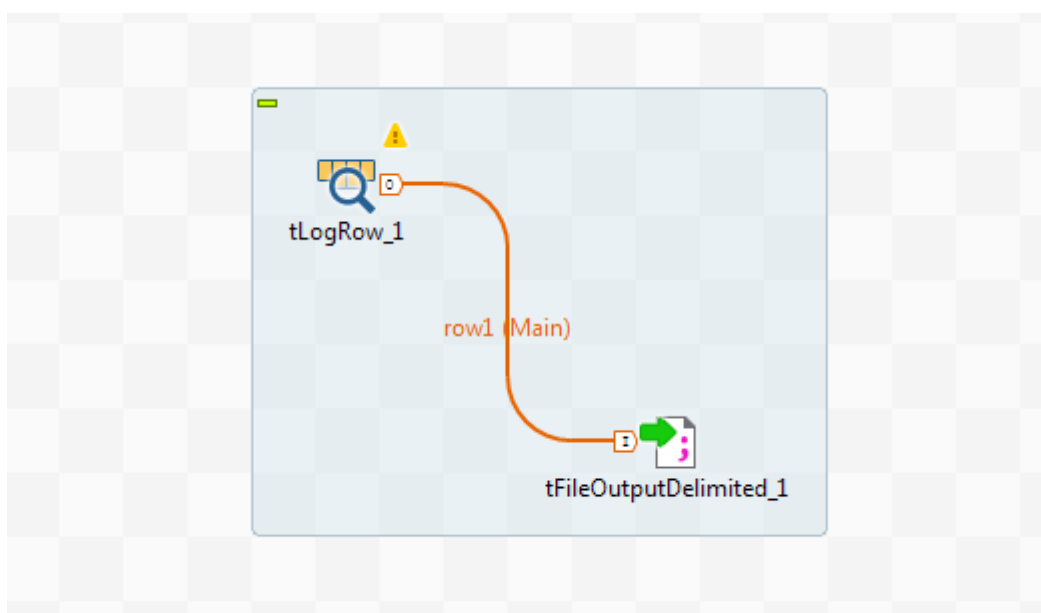
A text field and a component list appear. The component list shows all the components that can be connected with the input component.



- To narrow the search, type in the text field the name of the component you want to add or part of it, or a phrase describing the component's functionality if you don't know its name, and then double-click the component of interest, **tFileOutputDelimited** in this example, on the component list to add it onto the design workspace. The new component is automatically connected with the input component **tLogRow**, using a **Row > Main** connection.



To use a descriptive phrase as keywords for a fuzzy search, make sure the **Also search from Help when performing a component searching** check box is selected on the **Preferences > Palette Settings** view. For more information, see your *Talend Studio User Guide*.



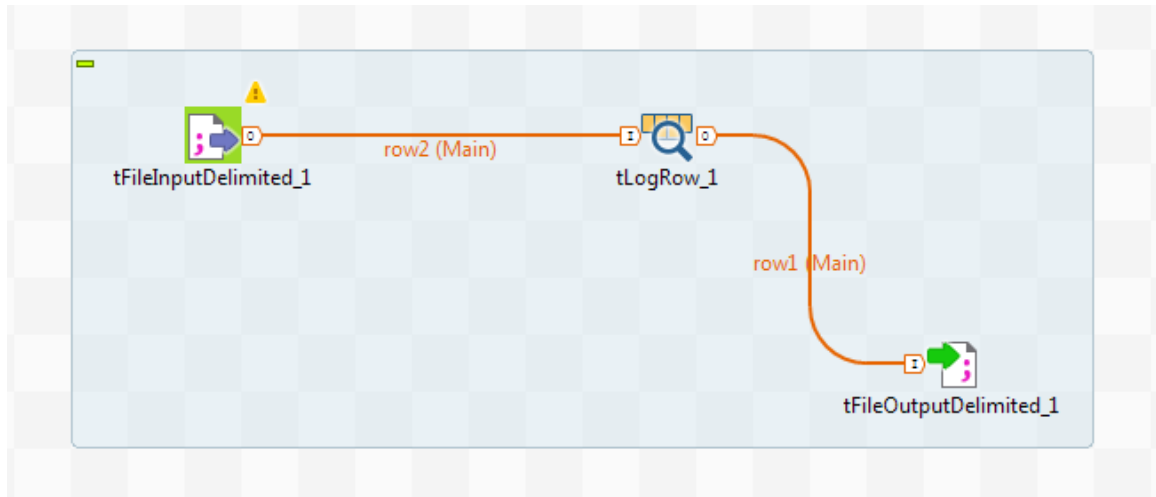
2.1.3. Connecting the components together

Now that the components have been added on the workspace, they have to be connected together. Components connected together form a subjob. Jobs are composed of one or several subjobs carrying out various processes.

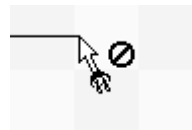
In this example, as the **tLogRow** and **tFileOutputDelimited** components are already connected, you only need to connect the **tFileInputDelimited** to the **tLogRow** component.

To connect the components together, proceed as follows:

1. Right-click the source component, **tFileInputDelimited** in this example.
2. In the contextual menu that opens, select the type of connection you want to use to link the components, **Row > Main** in this example.
3. Click the target component to create the link, **tLogRow** in this example.



Note that a black crossed circle is displayed if the target component is not compatible with the link.



According to the nature and the role of the components you want to link together, several types of link are available. Only the authorized connections are listed in the contextual menu.

2.1.4. Configuring the components

Now that the components are linked, their properties should be defined.

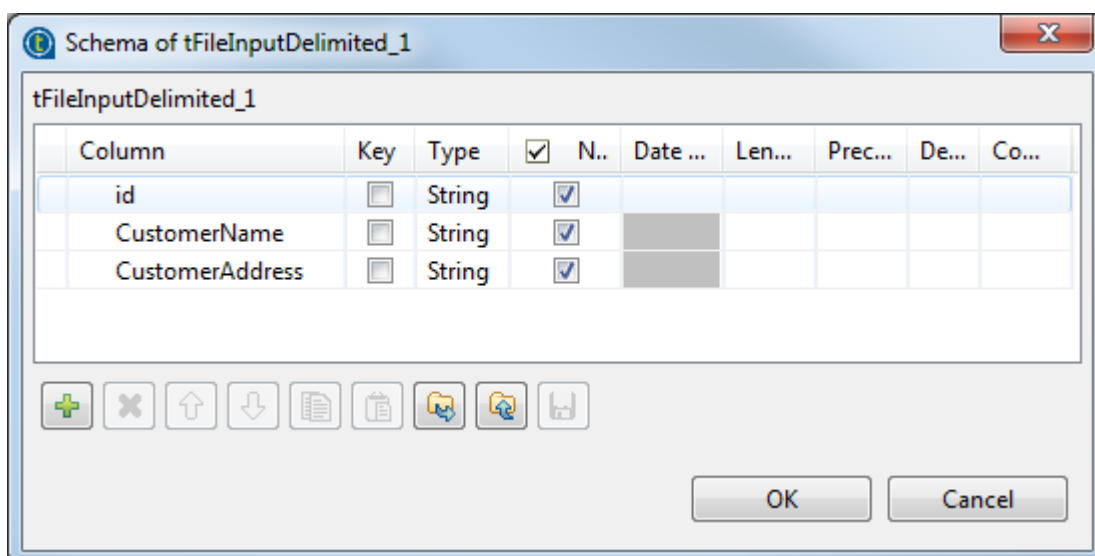
Configuring the tFileInputDelimited component

1. Double-click the **tFileInputDelimited** component to open its **Basic settings** view.

The screenshot shows the configuration window for the **tFileInputDelimited_1** component. The window has a sidebar on the left with tabs for **Basic settings**, **Advanced settings**, **Dynamic settings**, **View**, and **Documentation**. The **Basic settings** tab is active. The main area contains the following settings:

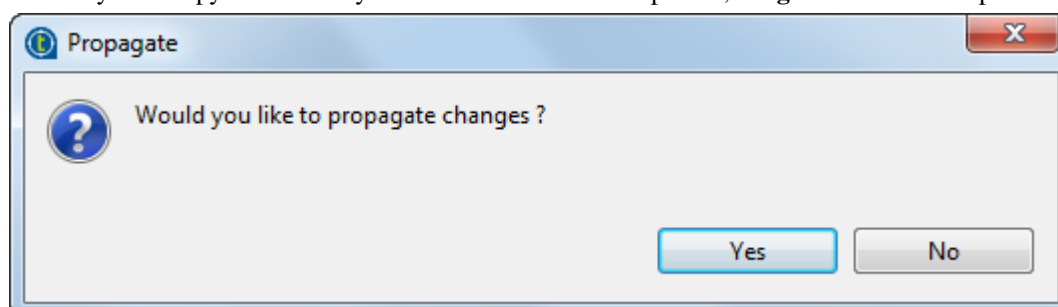
- Property Type:** Built-In (with a save icon)
- File name/Stream:** "C:/customers.txt" (with a browse icon)
- Row Separator:** "\n"
- Field Separator:** "," (with an asterisk)
- CSV options:** ☐ CSV options
- Header:** 1
- Footer:** 0
- Limit:** (empty field)
- Schema:** Built-In (with an **Edit schema** icon)
- Options:** ☒ Skip empty rows, ☐ Uncompress as zip file, ☐ Die on error

- Click the [...] button next to the **File Name/Stream** field.
- Browse your system or enter the path to the input file, *customers.txt* in this example.
- In the **Header** field, enter *I*.
- Click the [...] button next to **Edit schema**.
- In the Schema Editor that opens, click three times the [+] button to add three columns.
- Name the three columns *id*, *CustomerName* and *CustomerAddress* respectively and click **OK** to close the editor.



- In the pop-up that opens, click **OK** accept the propagation of the changes.

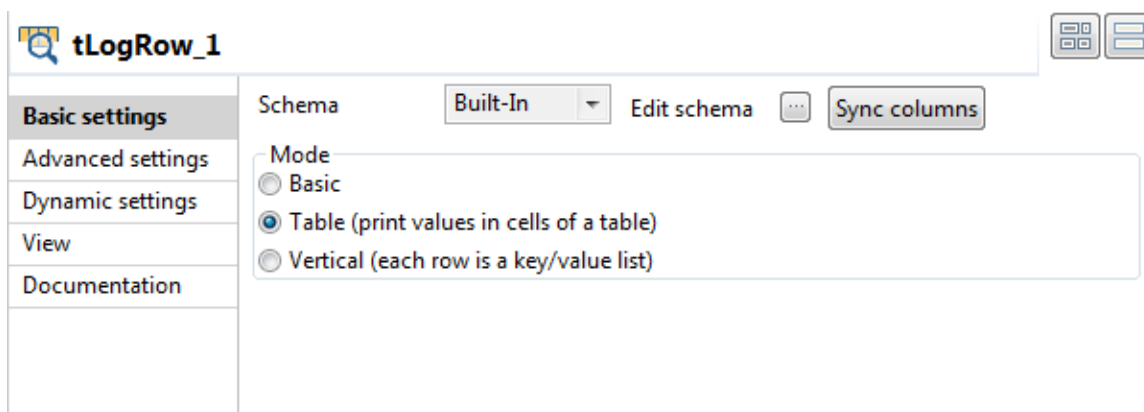
This allows you to copy the schema you created to the next component, **tLogRow** in this example.



Configuring the tLogRow component

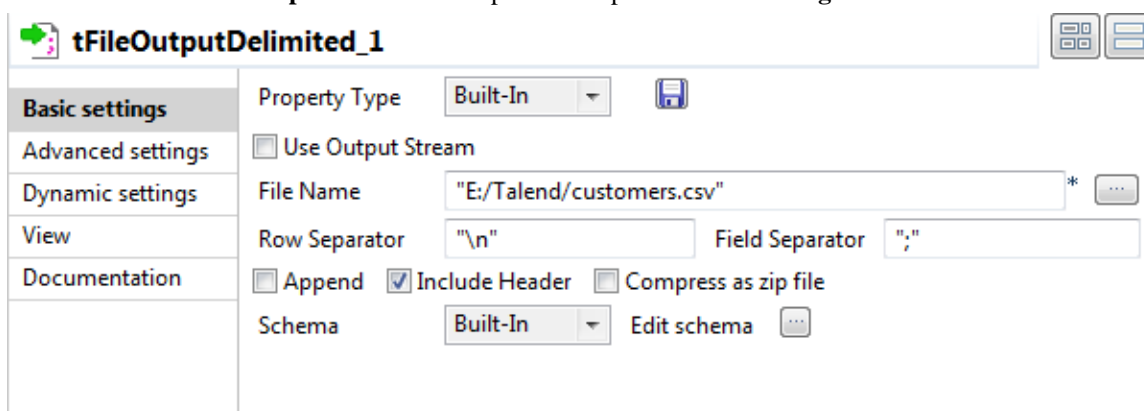
- Double-click the **tLogRow** component to open its **Basic settings** view.
- In the **Mode** area, select **Table (print values in cells of a table)**.

By doing so, the contents of the *customers.txt* file will be printed in a table and therefore more readable.



Configuring the tFileOutputDelimited component

1. Double-click the **tFileOutputDelimited** component to open its **Basic settings** view.



2. Click the [...] button next to the **File Name** field.
3. Browse your system or enter the path to the output file, *customers.csv* in this example.
4. Select the **Include Header** check box.
5. If needed, click the **Sync columns** button to retrieve the schema from the input component.

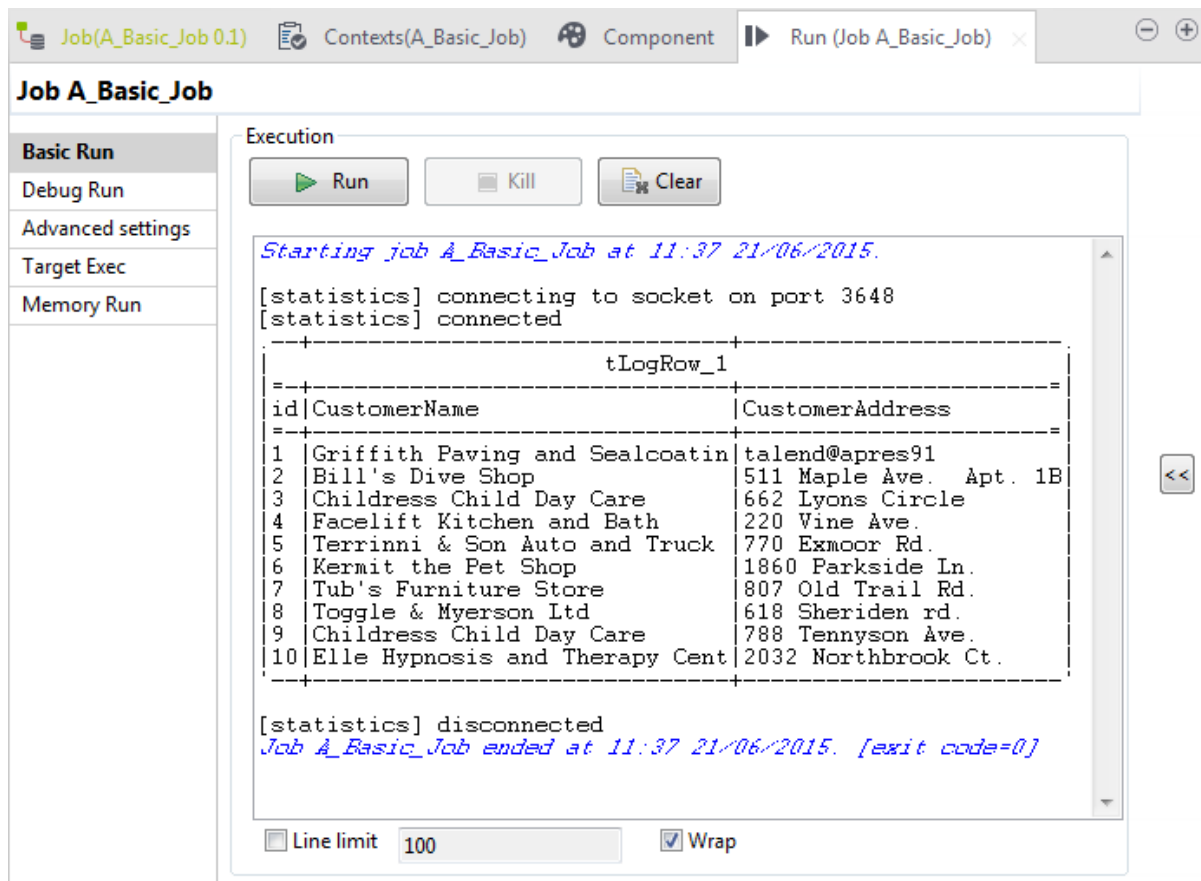
2.1.5. Executing the Job

Now that components are configured, the Job can be executed.

To do so, proceed as follows:

1. Press **Ctrl+S** to save the Job.
2. Go to **Run** tab, and click on **Run** to execute the Job.

The file is read row by row and the extracted fields are displayed on the **Run** console and written to the specified output file.



2.2. Use cases

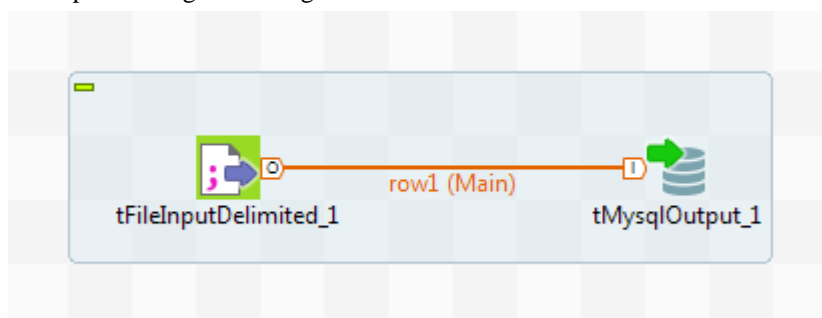
Let's get started with Job designs in *Talend Studio* by following a few basic end-to-end examples.

2.2.1. Updating data in a database table

This example describes a two-component Job that updates data in a MySQL table according to that in a delimited file.

Dropping and link components

1. Drop **tFileInputDelimited** and **tMysqlOutput** from the **Palette** onto the design workspace.
2. Connect the two components together using a **Row Main** link.



Configuring the input component

1. Double-click **tFileInputDelimited** to display its **Basic settings** view and define the component properties.
2. From the **Property Type** list, select **Repository** if you have already stored the metadata of the delimited file in the **Metadata** node in the **Repository** tree view. Otherwise, select **Built-In** to define manually the metadata of the delimited file.

For more information about storing metadata, see *Talend Studio User Guide*.

tFileInputDelimited_1

Basic settings

Property Type: Built-In

Advanced settings: "When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0."

Dynamic settings: File name/Stream: "D:/Java/Files/Input/customer_update.csv"

View: Row Separator: "\n" Field Separator: ","

Documentation: CSV options

Validation Rules: Header: 0 Footer: 0 Limit: 2000

Schema: Built-In Edit schema

☐ Skip empty rows ☐ Uncompress as zip file ☒ Die on error

3. In the **File Name** field, click the three-dot button and browse to the source delimited file that contains the modifications to propagate in the MySQL table.


In this example, we use the *customer_update* file that holds four columns: *id*, *CustomerName*, *CustomerAddress* and *idState*. Some of the data in these four columns is different from that in the MySQL table.

```
id;CustomerName;CustomerAddress;idState
858;Froggy's Gourmet Catering;1831 Beverly Place #9-11D;4
859;Dependable Plumbing and Sewver;1550 Ridge Rd.;25
860;Lickmen Restoration;1235 Easton Rd.;40
861;Acturial Enterprises Ltd.;3148 Cottonwood Ct.;18
862;Rythmics Ltd.;857 Woodbine Rd;30
863;Acturial Enterprises Ltd.;1482 Concorde Circle;48
864;Crosstracks Car Wash;218 Oakridge Ave.;39
865;Meonits & Mogogni Inc.; 616 Cobblestone Cir.;17
866;Foy Aviation;2220 Grant Blvd.;50
867;Ebert Music Center;12 Broadview Lane;29
868;janice Mann Accounting Service;1660 Park Ave.;9
869;Johnson, Erico & Co CPA's;2922 Twin Oaks Drive;40
870;Corbins;Rodriguez, & Savocchi;115 Pleasant Ave.;18
871;Nina's Snow Plowing;3385 University Ave.;20
872;Darcy Frame and Matting Servic;1101 Deerfield Place;47
873;Marks, Kaplan and Jones Ltd.;1949 Cloverdale Rd.;9
```

4. Define the row and field separators used in the source file in the corresponding fields.
5. If needed, set **Header**, **Footer** and **Limit**.

In this example, **Header** is set to 1 since the first row holds the names of columns, therefore it should be ignored. Also, the number of processed lines is limited to 2000.

6. Click the [...] button next to **Edit Schema** to open a dialog box where you can describe the data structure of the source delimited file that you want to pass to the component that follows.

Column	Key	Type	<input checked="" type="checkbox"/> Nullable
 id	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>
CustomerName	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>
CustomerAddress	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>
idState	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>

7. Select the **Key** check box(es) next to the column name(s) you want to define as key column(s).



It is necessary to define at least one column as a key column for the Job to be executed correctly. Otherwise, the Job is automatically interrupted and an error message displays on the console.

Configuring the output component

1. In the design workspace, double-click **tMySQLOutput** to open its **Basic settings** view where you can define its properties.

tMySQLOutput_1

Basic settings

Property Type: Built-In

DB Version: Mysql 5

☐ Use an existing connection

Host: localhost * Port: 3306 *

Database: test *

Username: root * Password: ***** *

Table: customers

Action on table: Default Action on data: Update

Schema: Built-In Edit schema Sync columns

Data source

This option only applies when deploying and running in the Talend Runtime

☐ Specify a data source alias

☐ Die on error

2. Click **Sync columns** to retrieve the schema of the preceding component. If needed, click the three-dot button next to **Edit schema** to open a dialog box where you can check the retrieved schema.
3. From the **Property Type** list, select **Repository** if you have already stored the connection metadata in the **Metadata** node in the **Repository** tree view. Otherwise, select **Built-In** to define manually the connection information in the corresponding fields: **Host**, **Port**, **Database**, **Username** and **Password**.

For more information about storing metadata, see *Talend Studio User Guide*.

4. In the **Table** field, enter the name of the table to update.
5. From the **Action on table** list, select the operation you want to perform, **Default** in this example since the table already exists.
6. From the **Action on data** list, select the operation you want to perform on the data, **Update** in this example.

Saving and executing the Job

1. Press **Ctrl+S** to save your Job.
2. Press **F6** or click **Run** on the **Run** tab to execute the Job.

id	CustomerName	CustomerAddress	idState
858	Froggy's Gourmet Catering	1831 Beverly Place #9D	4
859	Dependable Plumbing and Sewer	1550 Ridge Rd.	25
860	Lickmen Restoration	1235 Easton Rd.	40
id	CustomerName	CustomerAddress	idState
861	Acturial Enterprises Ltd.	3148 Cottonwood Ct.	18
862	Rythmics Ltd.	857 Woodbine Rd	30
863	Acturial Enterprises Ltd.	1482 Concorde Circle	48
864	Crosstracks Car Wash	218 Oakridge Ave.	39
865	Meonits & Mogogni Inc.	616 Cobblestone Cir.	17
866	Foy Aviation	2220 Grant Blvd.	50
867	Ebert Music Center	12 Broadview Lane	29
868	Janice Mann Accounting Service	1660 Park Ave.	9
869	Johnson, Erico & Co CPA's	2922 Twin Oaks Drive	40
870	Corbins, Rodriguez, & Savocchi	115 Pleasant Ave.	18
871	Nina's Snow Plowing	3385 University Ave.	20
872	Darcy Frame and Matting Serv	1101 Deerfield Place	47
873	Marks, Marks, and Kaplan Ltd.	1949 Cloverdale Rd.	9

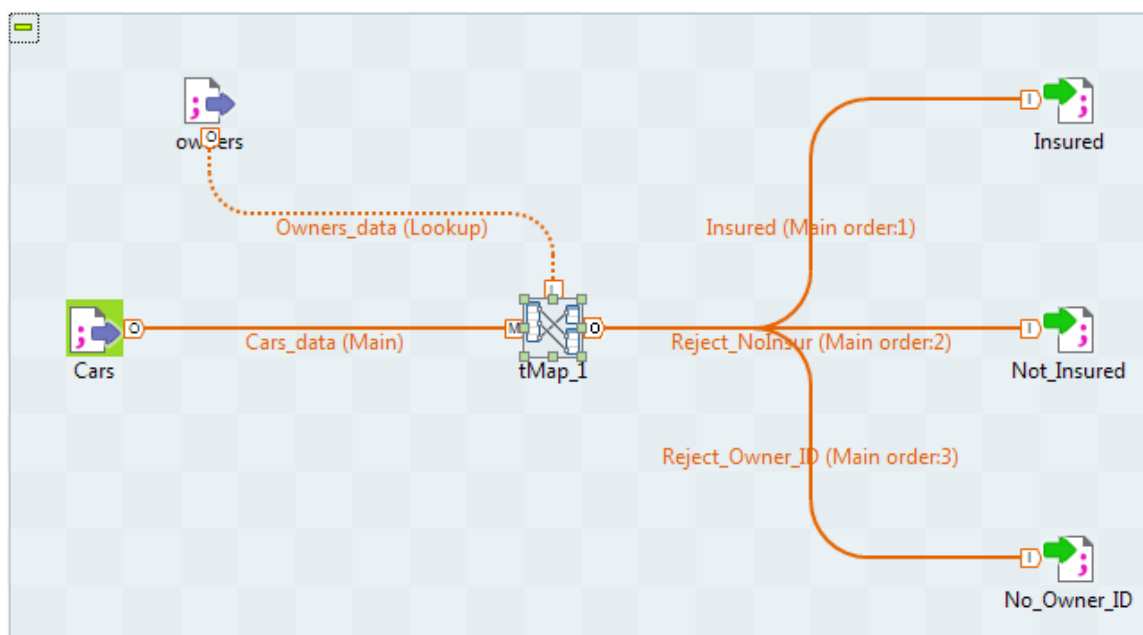
The MySQL table *customers* has been modified according to the delimited file.

2.2.2. Mapping data using a filter and a simple explicit join

The Job described below aims at reading data from a csv file, looking up at a reference file, and then extracting data from these two files based on a defined filter to an output file and reject files.

Adding and linking the components

1. Add two **tFileInputDelimited** components, a **tMap** and three **tFileOutputDelimited** components onto the design workspace.
2. Rename the two **tFileInputDelimited** components as *Cars* and *Owners*, either by double-clicking the label in the design workspace or via the **View** tab of the **Component** view.
3. Connect the two input components to **tMap** using **Row > Main** connections and label the connections as *Cars_data* and *Owners_data* respectively.
4. Connect **tMap** to the three output components using **Row > New Output (Main)** connections and name the output connections as *Insured*, *Reject_NoInsur* and *Reject_OwnerID* respectively.



Configuring the input components

1. Double-click the **tFileInputDelimited** component labelled *Cars* to display its **Basic settings** view.

2. Select **Repository** from the **Property type** list and select the component's schema, *cars* in this scenario, from the **[Repository Content]** dialog box. The rest fields are automatically filled.



This scenario assumes that the metadata of the input files is stored in the **Metadata** node of the **Repository** tree view for easy retrieval. For further information regarding metadata creation in the Repository, see *Talend Studio User Guide*.

If you do not have the metadata of your input files centralized in the **Repository**, you need to set the property type to **Built-In** and specify file path and define the file schema manually. Below is an abstract of the input file *cars.scv*:

```
ID_Owner;Registration;Make;Color;ID_Reseller
1;WZG 555;Ford;red;22
2;HYZ 472;Lexus;red;39
3;VYZ 862;Lexus;blue;21
4;ZYZ 350;Audi;red;31
5;EDZ 99;Audi;green;62
6;ZZX 845;Citroen;black;75
7;PBS 410;Renault;grey;11
8;JFO 929;Citroen;white;86
9;DPG 217;Lexus;black;13
```

- Double-click the component labelled *Owners* and repeat the setting operation. Select the appropriate metadata entry, *owners* in this scenario.

If you do not have the metadata of your input files centralized in the **Repository**, you need to set the property type to **Built-In** and specify file path and define the file schema manually. Below is an abstract of the reference input file *owners.csv*:

```
ID_Owner;Name;ID_Insurance;Chlidren_Nr
1;George EISENHOWER;108;8
2;James LINCOLN;35;8
3;William TAFT;6;10
4;Harry WILSON;134;3
5;Woodrow HOOVER;45;8
6;Chester TAYLOR;148;2
7;John REAGAN;31;8
8;Dwight POLK;105;7
9;William PIERCE;177;2
```

Configuring the mapping component

- Double-click the **tMap** component to open the **Map Editor**.

Note that the input area is already filled with the defined input tables and that the top table is the main input table, and the respective row connection labels are displayed on the top bar of the table.

- Create a join between the two tables on the *ID_Owner* column by simply dropping the *ID_Owner* column from the *Cars_data* table onto the *ID_Owner* column in the *Owners_data* table.
- Define this join as an inner join by clicking the **tMap settings** button, clicking in the **Value** field for **Join Model**, clicking the small button that appears in the field, and selecting **Inner Join** from the **[Options]** dialog box.

The screenshot shows the tMap configuration interface with two tables: **Cars_data** and **Owners_data**.

Cars_data table structure:

Column
ID_Owner
Registration
Make
Color
ID_Reseller

Owners_data table structure:

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Inner Join
Store temp data	false

Below the Owners_data table is a section for the join configuration:

Expr. key	Column
Cars_data.ID_Owner	ID_Owner
	Name
	ID_Insurance
	Children_Nr

A purple arrow indicates the join relationship between the *ID_Owner* column of *Cars_data* and the *ID_Owner* column of *Owners_data*.

- Drag all the columns of the *Cars_data* table to the *Insured* table.

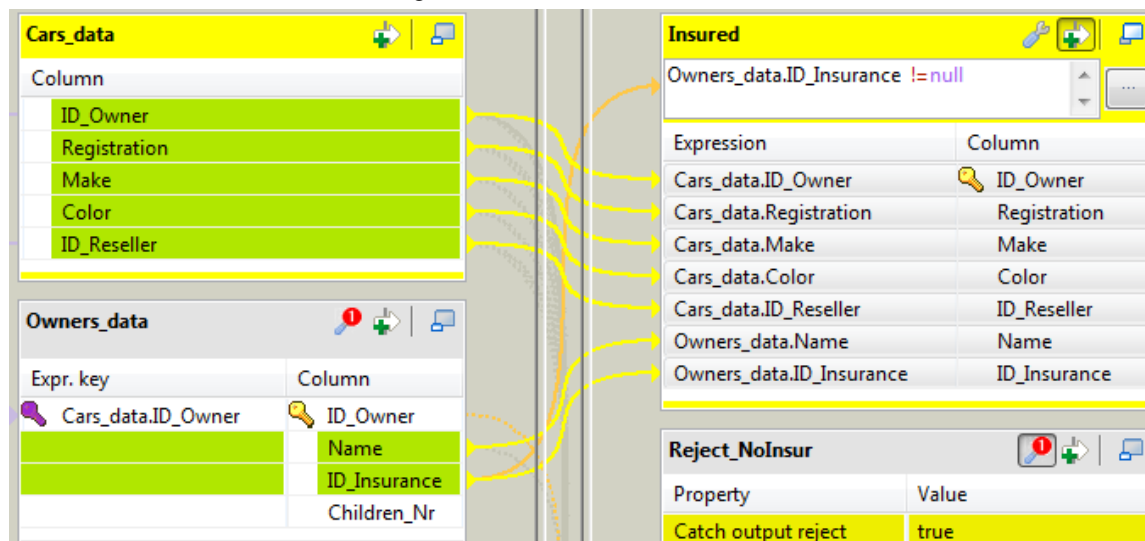
5. Drag the *ID_Owner*, *Registration*, and *ID_Reseller* columns of the *Cars_data* table and the *Name* column of the *Owners_data* table to the *Reject_NoInsur* table.
6. Drag all the columns of the *Cars_data* table to the *Reject_OwnerID* table.

For more information regarding data mapping, see *Talend Studio User Guide*.

7. Click the plus arrow button at the top of the *Insured* table to add a filter row.

Drag the *ID_Insurance* column of the *Owners_data* table to the filter condition area and enter the formula meaning 'not undefined': `Owners_data.ID_Insurance != null`.

With this filter, the *Insured* table will gather all the records that include an insurance ID.



8. Click the **tMap settings** button at the top of the *Reject_NoInsur* table and set **Catch output reject** to **true** to define the table as a standard reject output flow to gather the records that do not include an insurance ID.

Reject_NoInsur	
Property	Value
Catch output reject	true
Catch lookup inner join reject	false
Schema Type	Built-In
Expression	Column
Cars_data.ID_Owner	ID_Owner
Cars_data.Registration	Registration
Cars_data.ID_Reseller	ID_Reseller
Owners_data.Name	Name

9. Click the **tMap settings** button at the top of the *Reject_OwnerID* table and set **Catch lookup inner join reject** to **true** so that this output table will gather the records from the *Cars_data* flow with missing or unmatched owner IDs.

Reject_OwnerID	
Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Schema Type	Built-In
Expression	Column
Cars_data.ID_Owner	ID_Owner
Cars_data.Registration	Registration
Cars_data.Make	Make
Cars_data.Color	Color
Cars_data.ID_Reseller	ID_Reseller

Click **OK** to validate the mappings and close the **Map Editor**.

Configuring the output components

1. Double-click each of the output components, one after the other, to define their properties. If you want a new file to be created, browse to the destination output folder, and type in a file name including the extension.

Insured(tFileOutputDelimited_1)

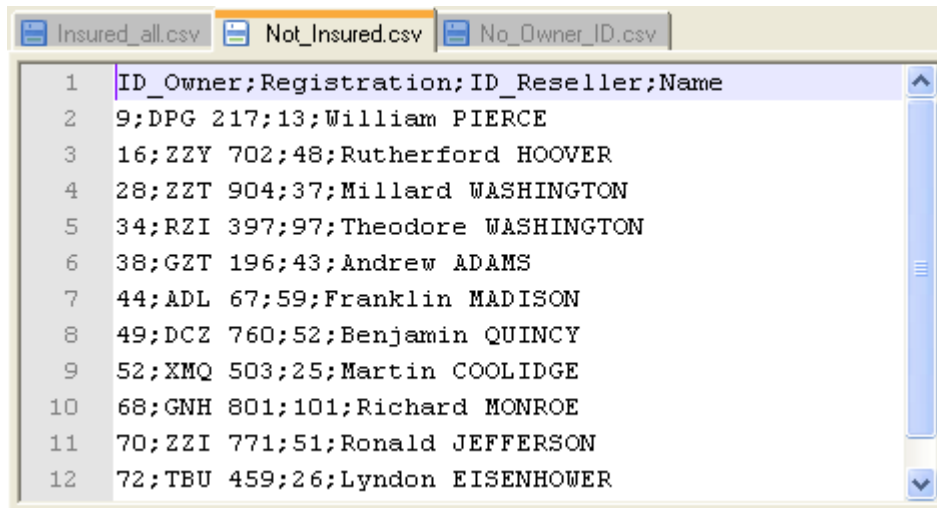
Basic settings	Property Type: Built-In
Advanced settings	<input type="checkbox"/> Use Output Stream
Dynamic settings	File Name: "C:/Output/Insured_all.csv"
View	Row Separator: "\n" Field Separator: ";"
Documentation	<input type="checkbox"/> Append <input checked="" type="checkbox"/> Include Header <input type="checkbox"/> Compress as zip file
Validation Rules	Schema: Built-In Edit schema Sync columns

2. Select the **Include header** check box to reuse the column labels from the schema as header row in the output file.

Executing the Job

1. Press **Ctrl + S** to save your Job.
2. Press **F6** to run the Job.

The output files are created, which contain the relevant data as defined.



	ID_Owner;Registration;ID_Reseller;Name
2	9;DPG 217;13;William PIERCE
3	16;ZZY 702;48;Rutherford HOOVER
4	28;ZZT 904;37;Millard WASHINGTON
5	34;RZI 397;97;Theodore WASHINGTON
6	38;GZT 196;43;Andrew ADAMS
7	44;ADL 67;59;Franklin MADISON
8	49;DCZ 760;52;Benjamin QUINCY
9	52;XMQ 503;25;Martin COOLIDGE
10	68;GNH 801;101;Richard MONROE
11	70;ZZI 771;51;Ronald JEFFERSON
12	72;TBU 459;26;Lyndon EISENHOWER



Chapter 3. Working in *Talend Studio* - basic Service and Route examples

This chapter will help you to get up and running with the Talend Studio by creating a simple SayHello example.

Here are the steps involved in the SayHello example:

1. Build a simple SayHello data service, in which a consumer sends a number of names to a service, which then prints "Hello!" to each of them in turn.
2. Build a simple SayHello route.



For more information about how to deploy the SayHello example into Talend Runtime, see the *Talend ESB Hands-on Guide*.

This section gives enough information to create and run the demo. For a comprehensive look at the Talend Studio User Interface, please see the *Talend Studio User Guide*.

For more details on specific components mentioned in this demo, please see the *Talend Components Reference Guide* for data services components and *Talend ESB Mediation Components Reference Guide* for mediation components.

3.1. Building a simple SayHello data service

There are a number of parts involved in creating this service, which you implement by dragging and dropping existing functionalities.

1. Create a SayHello provider.
2. Create a SayHello consumer.

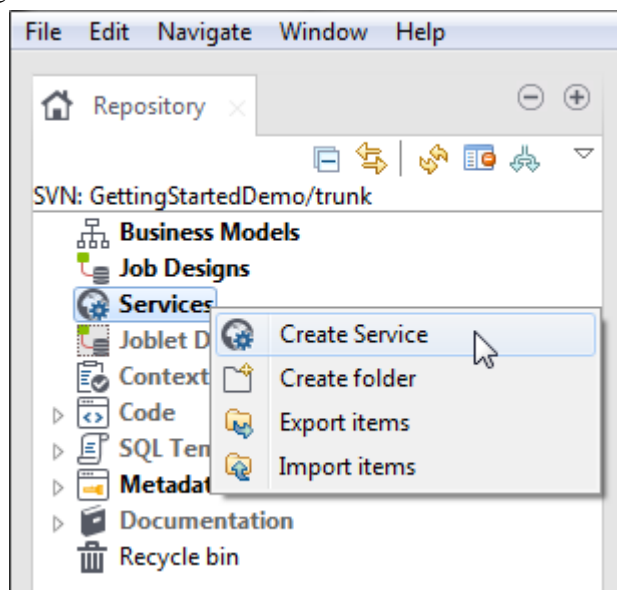
3.1.1. Creating a SayHello provider

This section provides you a step by step instruction to build the SayHello provider.

3.1.1.1. Creating a service

In this section, you will create a WSDL which defines the external contract to the Web service that clients can use.

1. To create the service, right-click **Services** in the left hand menu and select **Create Service**.



2. Enter the name *SayHelloService*, purpose *Demo* and a description of the service, and click **Next**.

New Service
Add a service in the repository

Name: SayHelloService

Purpose: demo

Description: A consumer sends a number of names to a service, which then prints "Hello!" to each of them in turn.

Author: jsmith@company.com

Locker:

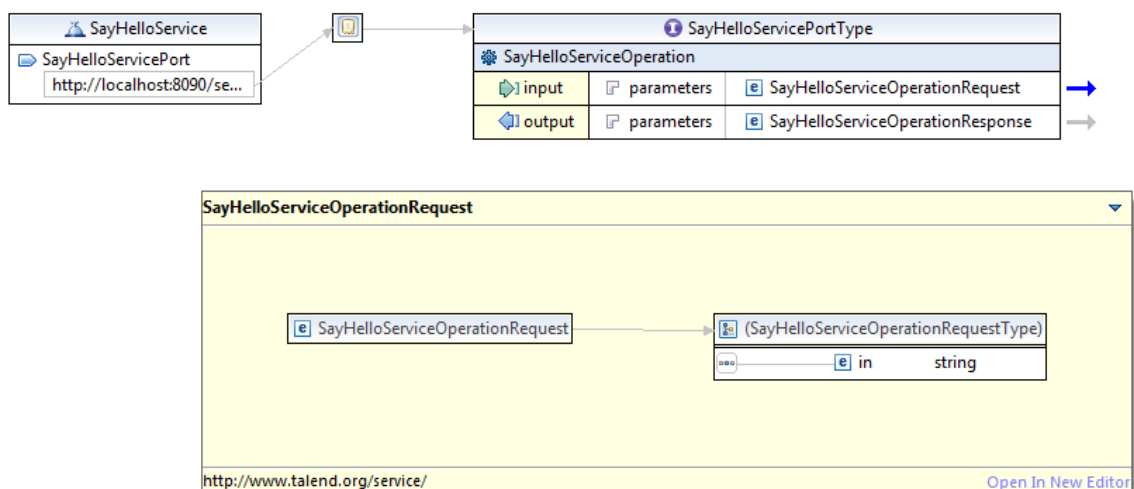
Version: 0.1 M m

Status:

Path: Select

< Back Next > Finish Cancel

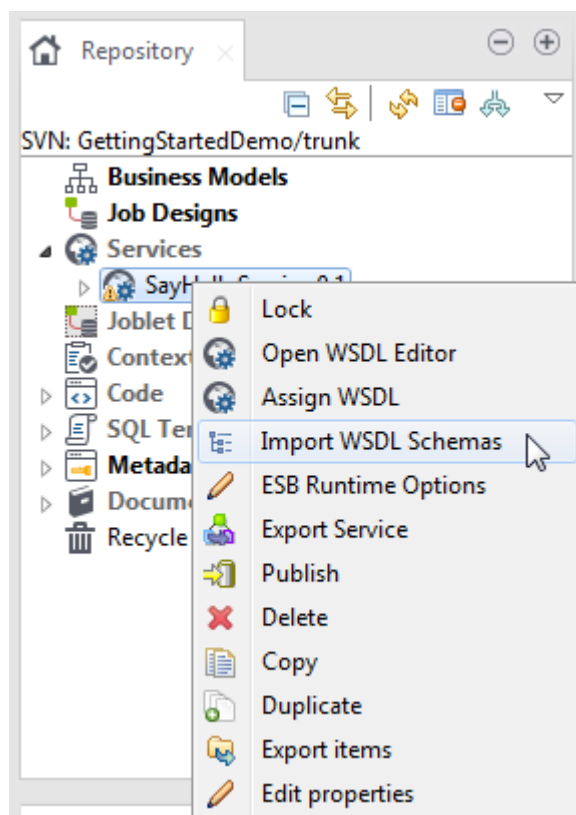
3. In the next step called **[Assign WSDL]**, select **Create new WSDL** and click **Finish** to return to the main screen.
4. Now the main window has a **[SayHelloService_0.1.wsdl]** tab displayed. This WSDL contains a new port (**SayHelloServicePort**) for the service, and default request and response operations (**SayHelloServiceOperationRequest** and **SayHelloServiceOperationResponse** respectively). Hover over the grey arrows to the right of the operations to display their parameters:



It is possible to make changes to the operations - add new operations and edit existing ones. However, the default operations are enough for this demo example.

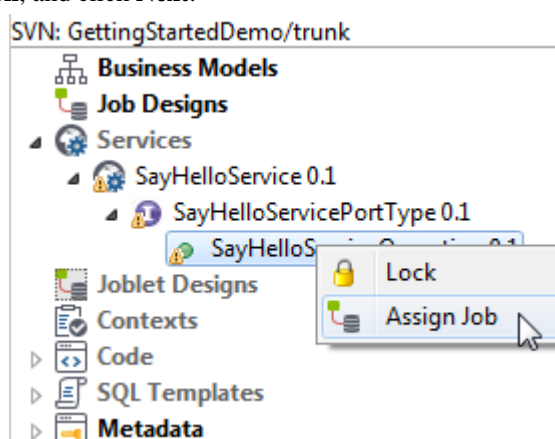
3.1.1.2. Configuring and exposing a service

1. Save the service details and WSDL Request / Response data types to the Metadata so that they can be accessible to other components. So, in **Services**, right-click **SayHelloService 0.1** and select **Import WSDL Schemas**.



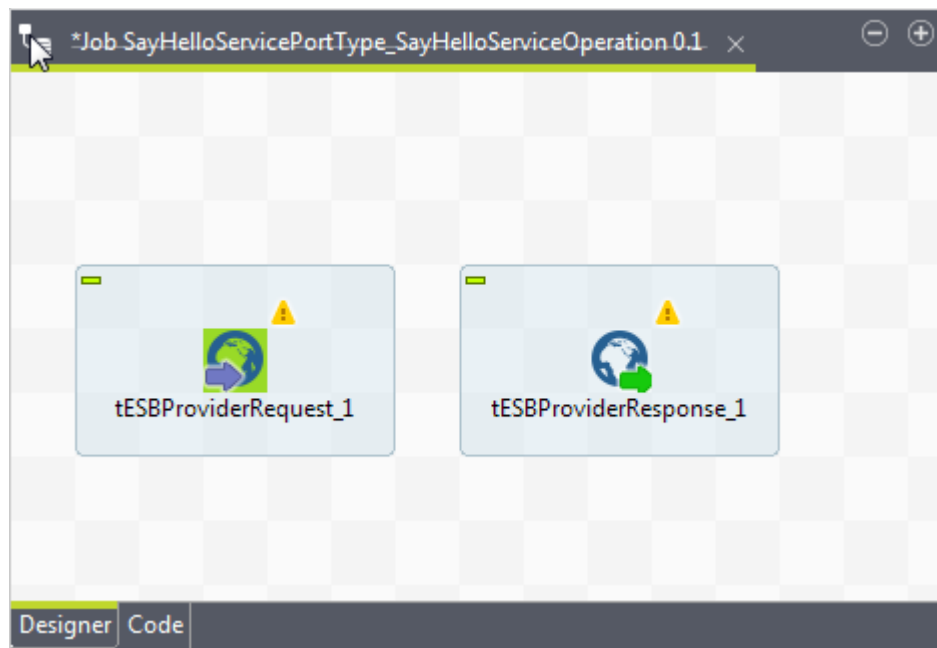
This option imports the WSDL metadata from the service into the **Repository**, under the **Metadata > File xml**. This allows you to share the operations details across services and other components.

2. Implement the operation - expand the elements displayed in **SayHelloService 0.1**, right-click **SayHelloServiceOperation 0.1** and select **Assign Job**. In the wizard, select **Create a new Job and Assign it to this Service Operation**, and click **Next**.

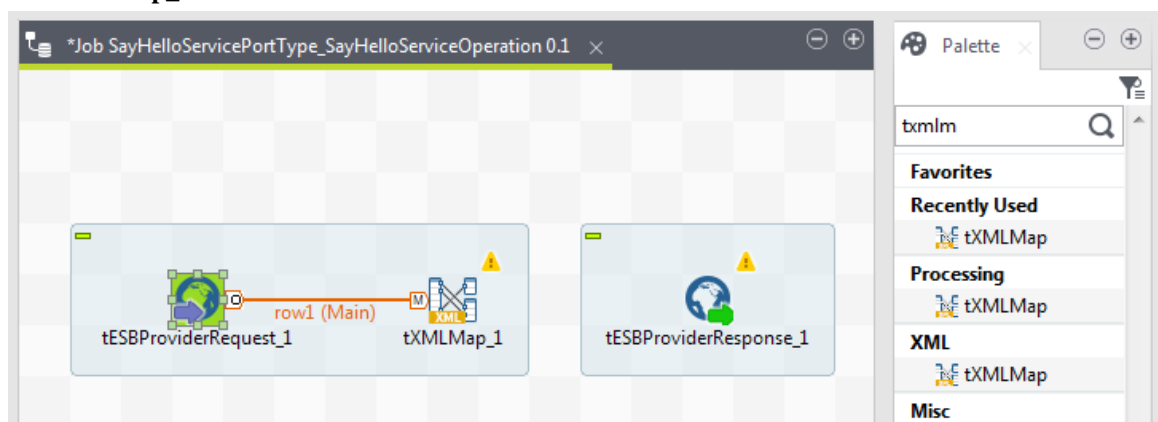


Keep the default name and prefix **SayHelloServicePortType_SayHelloServiceOperation**, and click **Finish**. This creates a new job, which allows to implement the operations using components in the main grid.

3. The default template of the **SayHelloServicePortType_SayHelloServiceOperation** Job is made of the **tESBProviderRequest** and the **tESBProviderResponse** components. Separate the two ESB components on the grid by clicking on the **tESBProviderResponse_1** icon, and dragging it further to the right.



- Now add some business logic. **tXMLMap** is a component that transforms and routes data from single or multiple sources to single or multiple destinations. Perform a search for the **tXMLMap** component in the Palette on the right hand side. There may be two instances found under different sections, but they are both the same, so select either. Drag and drop it between the two ESB components.
- Right-click the center of **tESBProviderRequest_1** and select **Row**, then **Main** and drop the end of the line on **tXMLMap_1**.

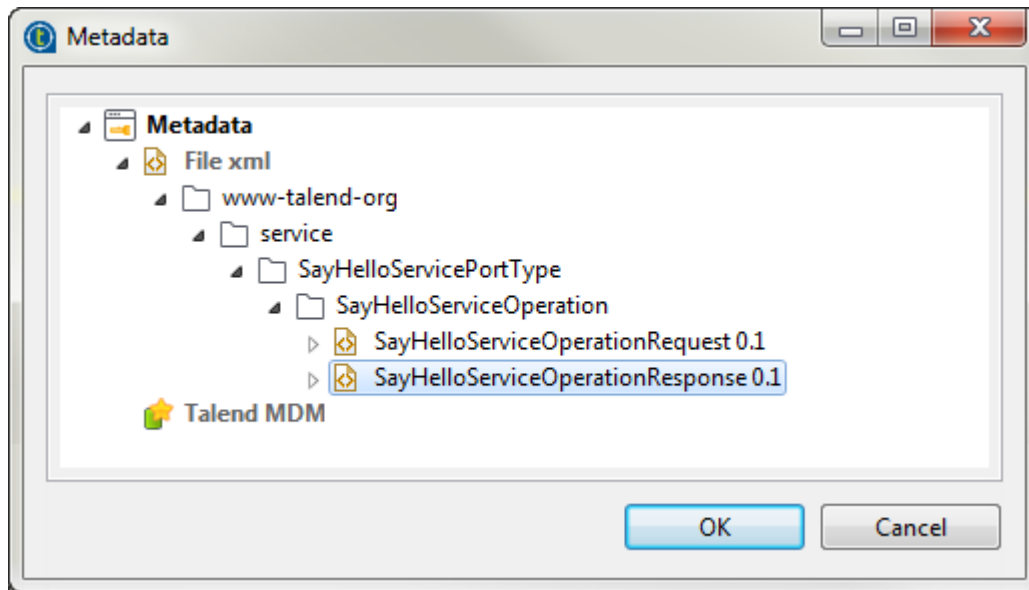


- Next, right-click **tXMLMap_1** and select **Row**, then **Main** and drop the end of the line on **tESBProviderResponse_1**. Give it the name *Response*, and click **OK**. Click the default **Yes** when asked if you wish to import the schemas.

3.1.1.3. Configuring the service operation

- Now customize the service operation to match the scenario. Double-click **tXMLMap_1** to open its editor. **tXMLMap** is used to route the information from the request to the response, and make use of the existing schema information from the WDSL.
- Under **main :row1** on the left hand side, right-click **payload** and select **Import from Repository**. In the **[Metadata]** wizard that appears, navigate from **File XML** to **SayHelloServiceOperationRequest0.1**, select it and click **OK**.

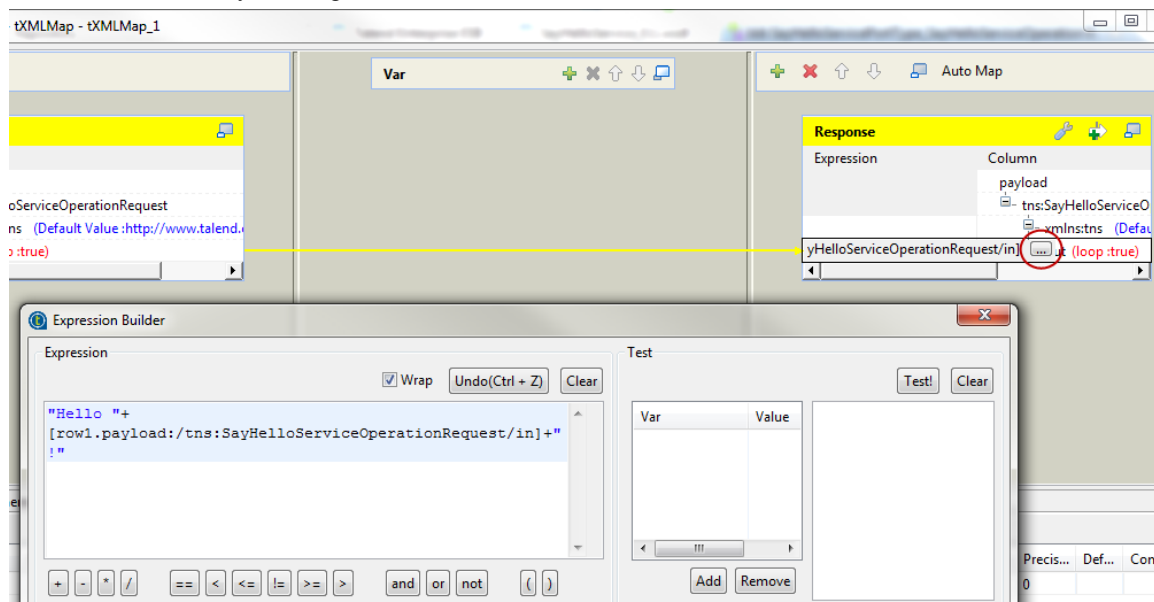
3. In the same way, on the right hand side, import the default response type, right-click **payload** then **Import from repository > File XML** and select **SayHelloServiceOperationResponse 0.1**.



Click **OK**.

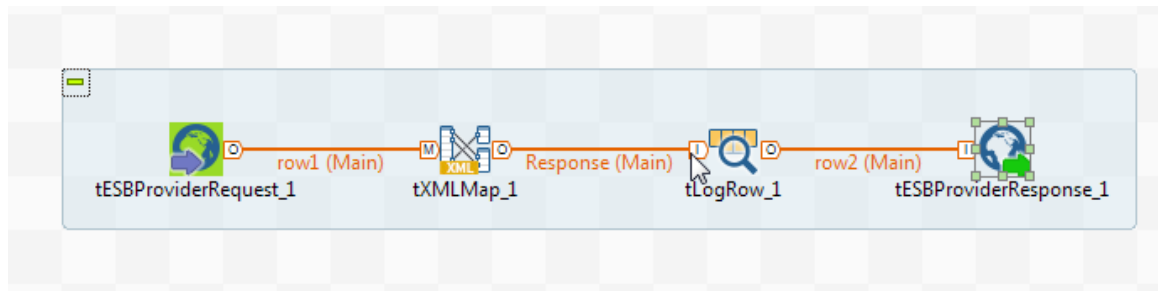
So, the request and response operations are implemented from the existing schemas.

4. Next, simply link the request schema to the response one: left-click **in** on the left hand side and drag it to the **out** expression in the response on the right hand side.
5. Next, modify the default expression that is sent. On the right hand side, under **Expression**, click the **HelloServiceOperationRequest** value, and click the [...] button beside it. Edit the expression (which will evaluate to a name) by clicking in the field, and add "Hello " + before it, and +"!" after it.



Click **OK**. You will see the updated expression now on the right hand side. Click **OK** to return to the main Job design window.

6. Finally, in order to see more as the Job executes, add a logging component. This is done simply by searching for **tLog** in the palette, and dragging **tLogRow** from the Palette on the right hand side and dropping it on the **Response** link between the **tXMLMap_1** and **tESBProviderResponse_1**.



Now, the implementation of the **SayHelloServiceOperation** is complete.

3.1.1.4. Running the service in the Talend Studio

Do a quick check now to make sure this part is working, by clicking the tab **Run (Job SayHello...)** in middle section in the bottom half of the window, and then clicking the **Run** button.

Job SayHelloServicePortType_SayHelloServiceOperation

Basic Run

Debug Run

Advanced settings

Target Exec

Memory Run

Execution

Run Kill Clear

```

Starting job
SayHelloServicePortType_SayHelloServiceOperation at
18:51 26/06/2015.

[statistics] connecting to socket on port 4027
[statistics] connected
Jun 26, 2015 6:51:51 PM
org.apache.cxf.wsdl.service.factory.ReflectionServiceFa
ctoryBean buildServiceFromWSDL
INFO: Creating Service
{http://www.talend.org/service/}SayHelloService from
WSDL:
C:/600NB/TOS_ESB-20150625_1935-V6.0.0SNAPSHOT/workspace/
GETTINGSTARTEDDEMO/services/SayHelloService_0.1.wsdl
Jun 26, 2015 6:51:51 PM
org.apache.cxf.endpoint.ServerImpl initDestination
INFO: Setting the server's publish address to be
http://localhost:8090/services/SayHelloService
2015-06-26
18:51:51.651:INFO:oejs.Server:jetty-8.1.14.v20131031
2015-06-26
18:51:51.679:INFO:oejs.AbstractConnector:Started
SelectChannelConnector@localhost:8090
web service [endpoint:
http://localhost:8090/services/SayHelloService]
published
  
```

☐ Line limit 100 ☒ Wrap

The Job is compiled, and the log output shows that the Web service has been assigned a port 8090 and has been published for other services to use.

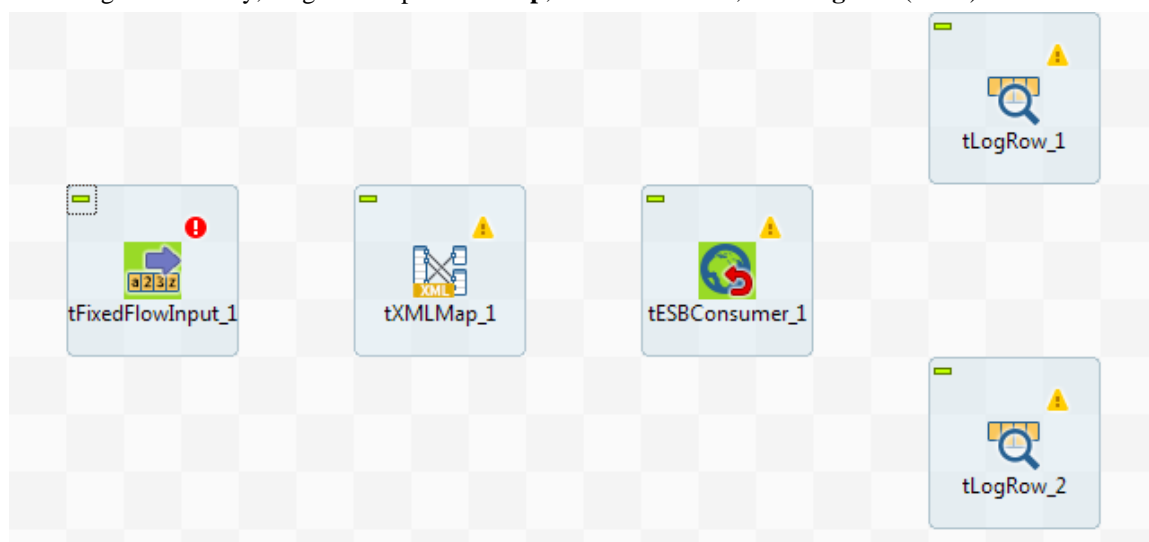
Select and copy "http://localhost:8090/services/SayHelloService" for later use.

3.1.2. SayHello consumer

In this section, you will see how to create the SayHello consumer, and call the service with it.

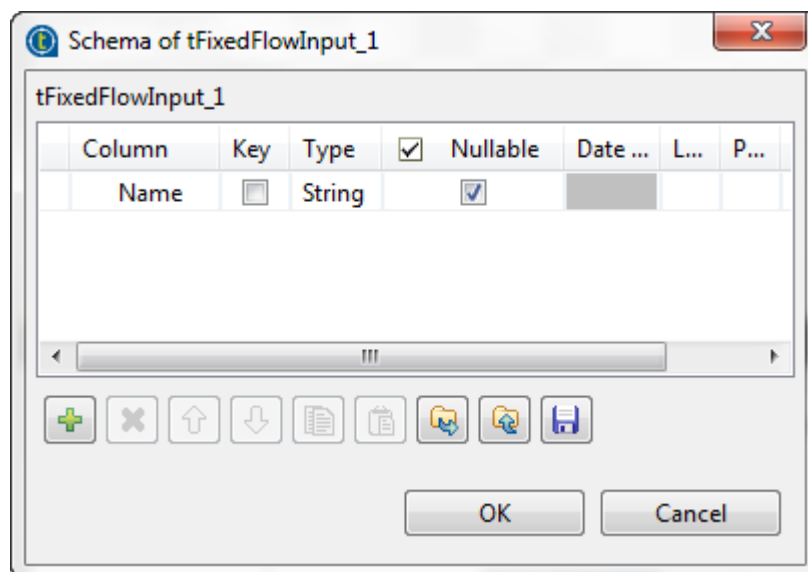
3.1.2.1. Creating the SayHello consumer

1. To test the service, you can also create a small consumer Job. Right-click **Job Designs** and select **Create Job**. In the **Name** field, type *SayHelloConsumer*, and the purpose is *Demo*. Click **Finish**.
2. Now in the **[Job SayHelloConsumer 0.1]** tab, search for **tFixedFlowInput** in the palette, and drag and drop it onto the grid. Similarly, drag and drop **tXMLMap**, **tESBConsumer**, and **tLogRow** (twice) as shown below.



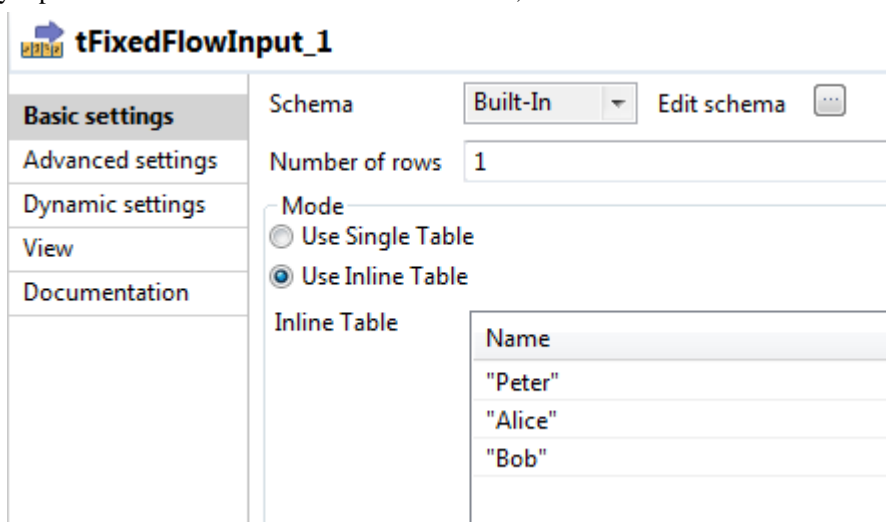
The **tFixedFlowInput** components generates as many lines and columns as you want using context variables, and the **tESBConsumer** calls a specified method from the invoked Web service, and returns a class, based on parameters.

3. Now configure the components. Double-click the center of **tFixedFlowInput_1** and in **Component** tab below, select **Use Inline Table**. Then click the [...] button next to **Edit schema** to open the Schema editing window.



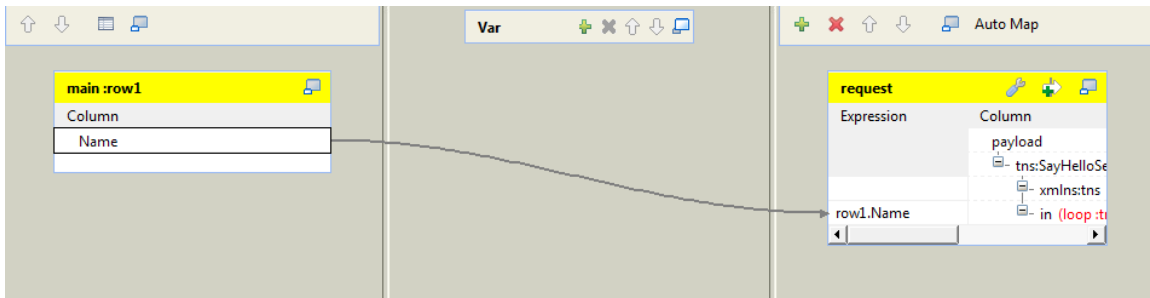
In this window, click [+] to add a string argument, and rename **newColumn** to **Name** and click **OK** to close this window.

4. Returning to the **tFixedFlowInput_1 Component** tab, use the [+] button to add sample rows, and successively replace the "newLine" text with names "Peter", "Alice" and "Bob".

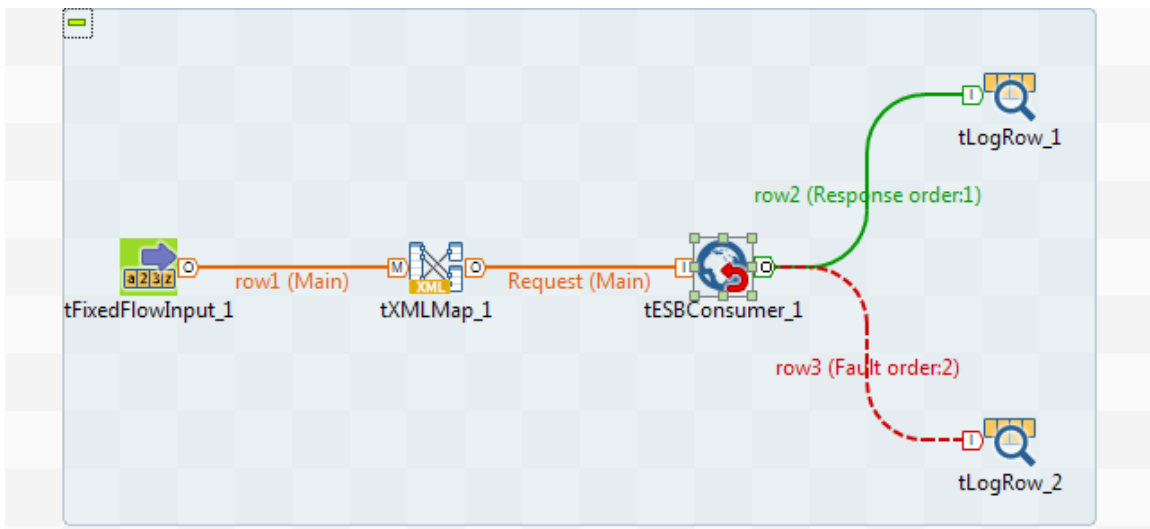


This is the example data that the consumer will send to the *SayHelloService*.

5. Now, in the **Job SayHelloConsumer 0.1** Job, link the components as before, by right-clicking and dragging **tFixedFlowInput_1** > **row** > **main** to **tXMLMap_1**. Then right-click **tXMLMap_1** > **row** and name this new Output *Request* (click the default **yes** to get the schema of the target component), and drop the end on **tESBConsumer_1**.
6. Now double-click the **tXMLMap** to configure it as before. On the right hand side, click **payload** > **Import From Repository**. In the **[Metadata]** wizard that appears navigate from **File XML** to **SayHelloServiceOperationRequest 0.1**, and select it. This enables you to call the service operation. Click **OK**.
7. Now left-click and drag **Name** on the left hand side to the **in** parameter on the right hand side and click **OK** to return to the main window.



- Finally, take care of the response outputs. Right-click the center of **tESBConsumer_1**, drag and select **Row > Response** and drop the end on **tLogRow_1** so that any responses should go there. Similarly, right-click **tESBConsumer_1**, select **Row > fault** and drop the end on **tLogRow_2** so that any faults should go there.



Now, the consumer Job configuration is complete.

In summary, **tFixedFlowInput** generates "Peter", "Alice", "Bob", these will be passed by **tXMLMap** to **tESBConsumer**, which will do three corresponding invocations on the target provider.

3.1.2.2. Running the consumer

- Now, point the consumer at the correct WSDL endpoint for the Service. So go to the **Component** tab of the **tESBConsumer** component, and edit the endpoint there to reference the correct service. Click the [...] button next to **Service Configuration**, and the WSDL configuration window opens. Paste in "http://localhost:8090/services/SayHelloService" to replace the service address there, giving a full address of "http://localhost:8090/services/SayHelloService?WSDL" and click the refresh button to the right to load the information. Click **Finish**.

Configure component with Web Service operation

WSDL:

Port Name:

Operation:

☐ Populate schema to repository on finish

- Finally, run the consumer Job. Click the **Run (Job SayHelloConsumer)** tab, and click **Run**. The Job builds and executes, and the three names in a Hello message display in the output.

Job SayHelloConsumer

Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run

Execution

```

Starting job SayHelloConsumer at 18:59 26/06/2015.

[statistics] connecting to socket on port 3581
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<tns:SayHelloServiceOperationResponse
xmlns:tns="http://www.talend.org/service/"><out>Hello
Peter!</out></tns:SayHelloServiceOperationResponse>
<?xml version="1.0" encoding="UTF-8"?>
<tns:SayHelloServiceOperationResponse
xmlns:tns="http://www.talend.org/service/"><out>Hello
Alice!</out></tns:SayHelloServiceOperationResponse>
<?xml version="1.0" encoding="UTF-8"?>
<tns:SayHelloServiceOperationResponse
xmlns:tns="http://www.talend.org/service/"><out>Hello
Bob!</out></tns:SayHelloServiceOperationResponse>
[statistics] disconnected
Job SayHelloConsumer ended at 18:59 26/06/2015. [exit
code=0]

```

So, the creation and execution of a SayHello consumer and provider in Talend Studio is successful.

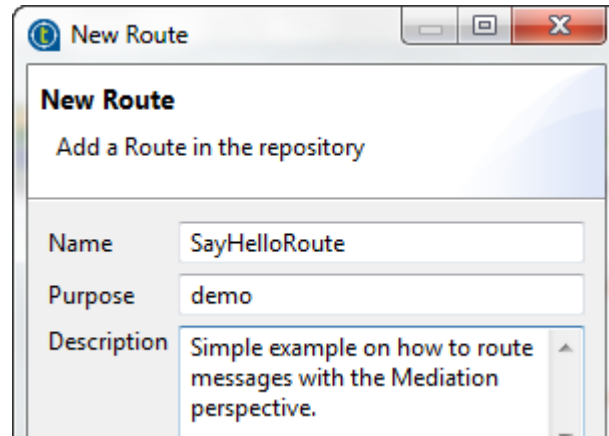
3.2. SayHelloRoute example

In this example, you will see how to extend your existing SayHello consumer and provider. Using the **Mediation** perspective, you will build a route that filters the Hello messages by name, so that messages with the name "Alice" in them go to the service provider, and other names will get error messages.

Finally, you will run the consumer from the first example, and show the messages coming from the consumer, and being routed to the correct endpoint.

3.2.1. Creating the route

1. First, switch from the **Integration** perspective to the **Mediation** perspective by clicking **Mediation** in the top right hand corner.
2. Create a new route by right-clicking **Routes > Create Route**. Give the name *SayHelloRoute* and purpose is *Demo*, and click **Finish**.



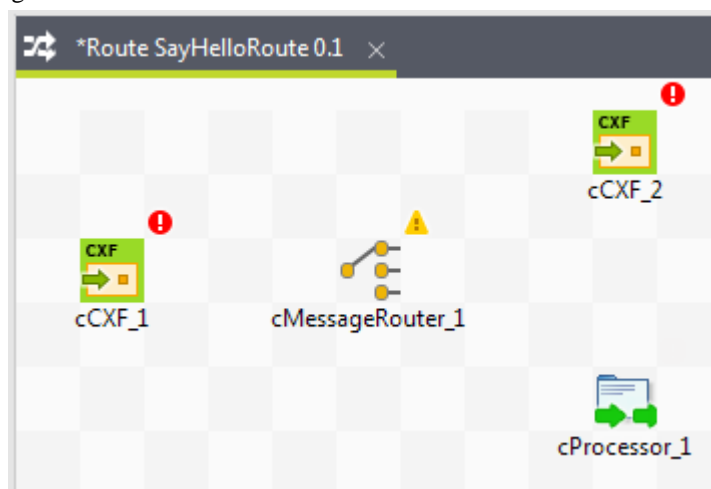
The **Route SayHelloRoute 0.1** tab opens.

3. Now you will notice that the palette has changed from the one in the **Integration** perspective to the **Mediation** perspective. You will create a typical content-based route, dragging and dropping components from the palette to the route grid.

The request message is coming in from the consumer, so drag and drop the **Connectivity > Services > cCXF** component, which intercepts messages coming into server endpoints.

Then, as you want to create a content-based route, drag and drop the **Routing > cMessageRouter** component, which reroutes messages depending on a set of conditions.

Then add one more **Connectivity > Services > cCXF** for the target service, and a **Custom > cProcessor** to return error messages.



Having multiple **cCXF** components with the same label in a Route is not supported.

It is recommended to label each component with a unique name to better identify its role in the Route.

Having duplicate labels may cause problems for the code generation of some components.

4. To implement this, just add some parameters. So, click **cCXF_1**, and then click the **Component** tab below:
 - As a Talend Open Studio for ESB first time user, you will be asked to install external libraries in the Studio to be able to use some components as the **cCXF**, so please follow the instructions and install them.
 - In the **Address** field, paste in the previous service address "http://localhost:8090/services/SayHelloService", and update the port to be 8092, since the new service will be listening on this port. So, you get the following new service address "http://localhost:8092/services/SayHelloService".
 - In the **WsdI File** field, specify the URL of the WSDL from the original service. Use the http:// address to get the live background service information. It is in "http://localhost:8090/services/SayHelloService?WSDL".
 - In the **Dataformat** list, select **PAYLOAD**, as you are looking at the message body.



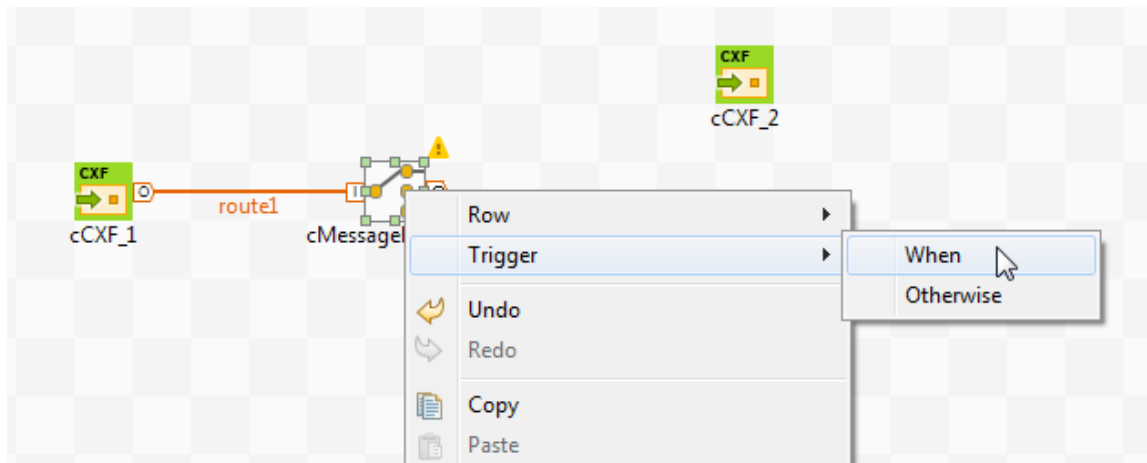
At any point, you can save the current information by selecting **File > Save** or clicking the **Save** icon in the top left hand corner.

5. Repeat the previous step for the **cCXF_2**: - except that the port number in the **Address** field is 8090.
6. Configure the **cProcessor** component to return error messages. Double-click it in the design workspace to show its **Basic settings** view in the **Component** tab.

In the **Code** box, enter `throw new Exception("user name error")` to throw an exception.



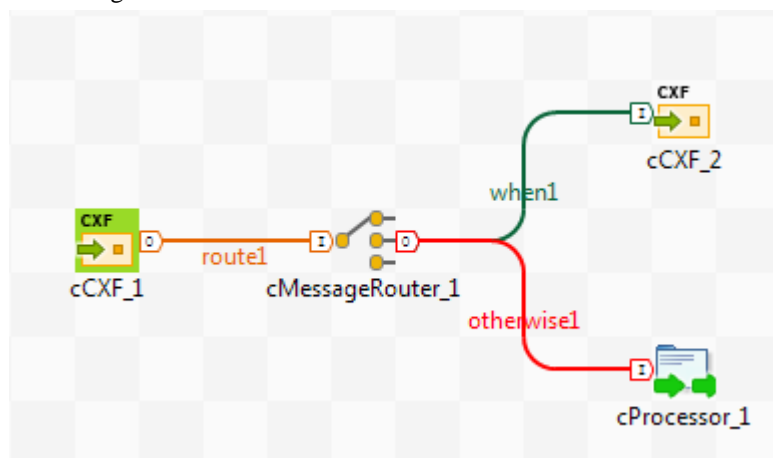
7. So now, connect the components together. Connect **cCXF_1** to **cMessageRouter_1** by right-clicking the center of **cCXF_1** and selecting **Row > Route** and dropping the end onto **cMessageRouter_1**.
8. Then create a **When** trigger for the service, by right-clicking **cMessageRouter_1**, selecting **Trigger > When** and dropping the end on **cCXF_2**.



9. Similarly, add an **Otherwise** trigger from **cMessageRouter_1** to **cProcessor_1** by right-clicking **cMessageRouter_1**, selecting **Trigger > Otherwise** and dropping the end on **cProcessor_1**.

So, in summary, you create a request on port 8092 (**cCXF_1**), and send it to either port 8090 (**cCXF_2**) or the **cProcessor** endpoint, depending on the contents of the message.

This results in the following:



10. Configure the **When** condition: right-click the **when1** line, which brings up a small dialog.

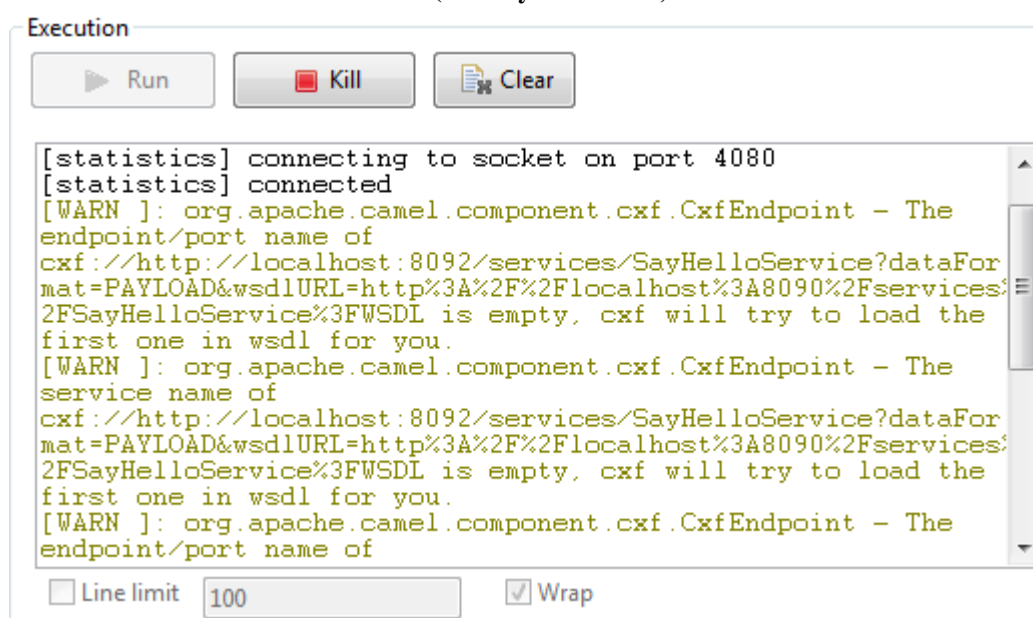
In the **Type** list, select **simple**, and the **Condition** is `"${bodyAs(String)} contains 'Alice'"`.

Type	simple	<input type="checkbox"/> Append endChoice()
Condition	<code>"\${bodyAs(String)} contains 'Alice'"</code>	

This way, any message with "Alice" in the body will be routed to the service that listens on port 8090.

3.2.2. Running the services

1. To test that all has been configured correctly, before adding the consumer, go to the route created in the studio and execute it. Click the **Run** button in **Run (Job SayHelloRoute)** tab:

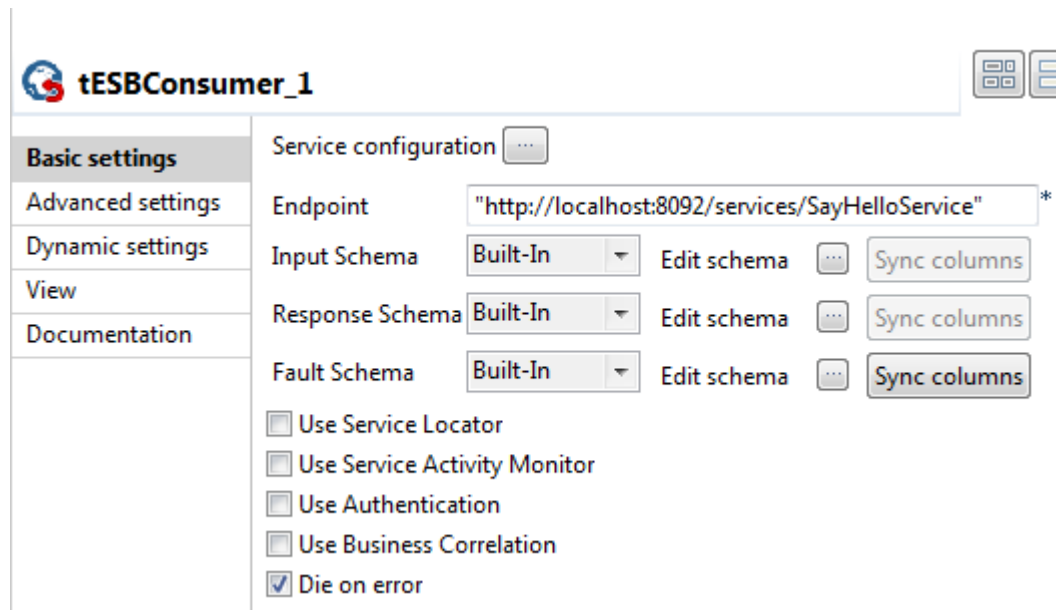


This checks that the CXF configuration information is syntactically correct. It also gives an output of "0 rows" on the grid, which reflect the flow of messages, so that you know that the parts are connecting. Leave the job running.



If you get syntax errors, then click on each component in turn, and examine it in the **Component** tab. Check in particular that the double quotes are all there, and that the port numbers are correct.

2. Now, switch back to the consumer to run the demo for real. Click **Integration** perspective in the top right hand corner. Then open the **SayHelloConsumer 0.1**. Update the port number, so click the **Component** tab of the **tESBConsumer**, and the [...] button next to **Service Configuration**. Update the port number to be 8092, and click the refresh button to retrieve the WSDL information. Now update the endpoint to be that of the route - `http://localhost:8092/services/SayHelloService`.



tESBConsumer_1

Basic settings

Advanced settings

Dynamic settings

View

Documentation

Service configuration

Endpoint: "http://localhost:8092/services/SayHelloService" *

Input Schema: Built-In Edit schema Sync columns

Response Schema: Built-In Edit schema Sync columns

Fault Schema: Built-In Edit schema Sync columns

☐ Use Service Locator

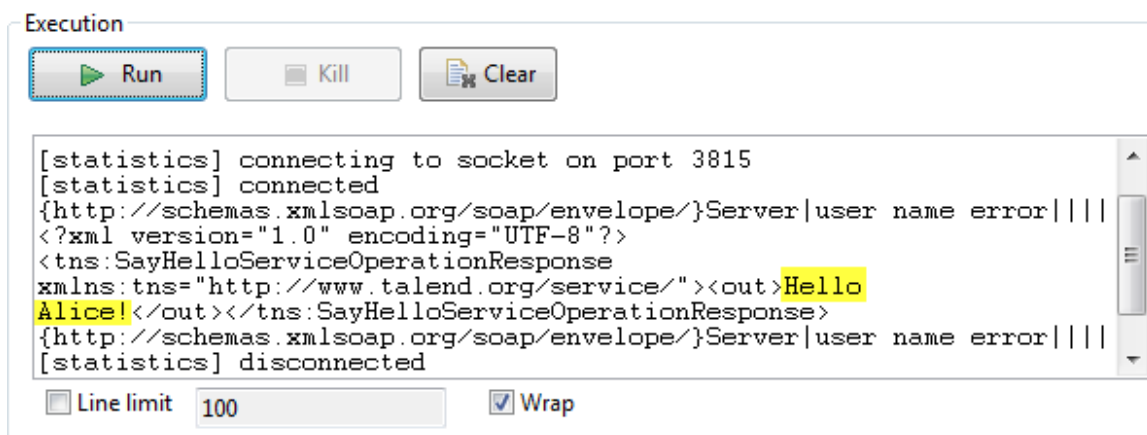
☐ Use Service Activity Monitor

☐ Use Authentication

☐ Use Business Correlation

☒ Die on error

- Now send a request by running the consumer job by clicking on the tab **Run (Job SayHelloConsumer)** and click **Run**. The "Hello Alice" and two error messages are displayed in the consumer output.



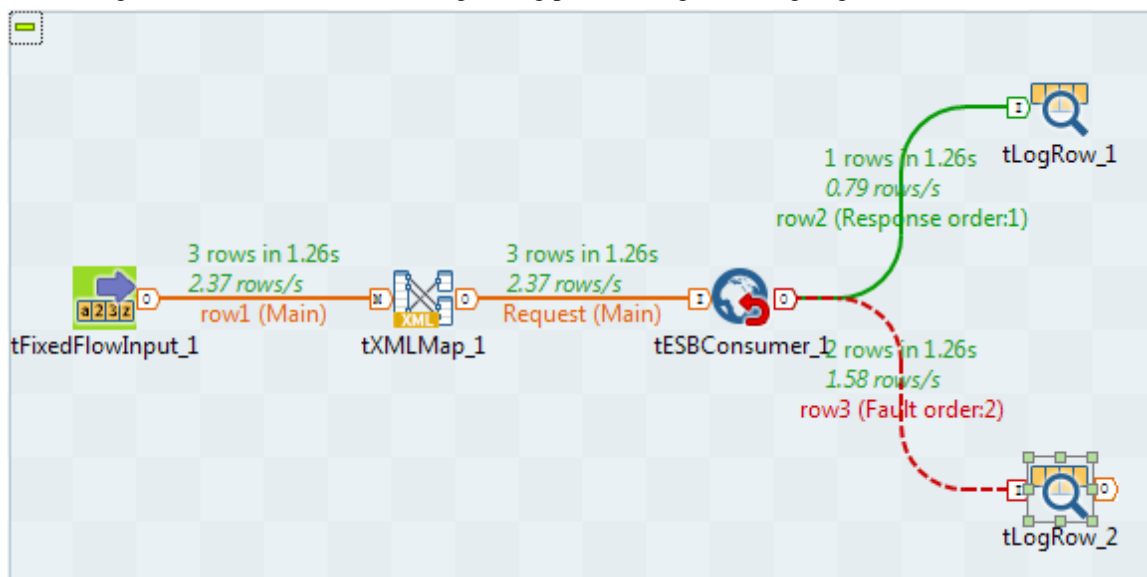
Execution

Run Kill Clear

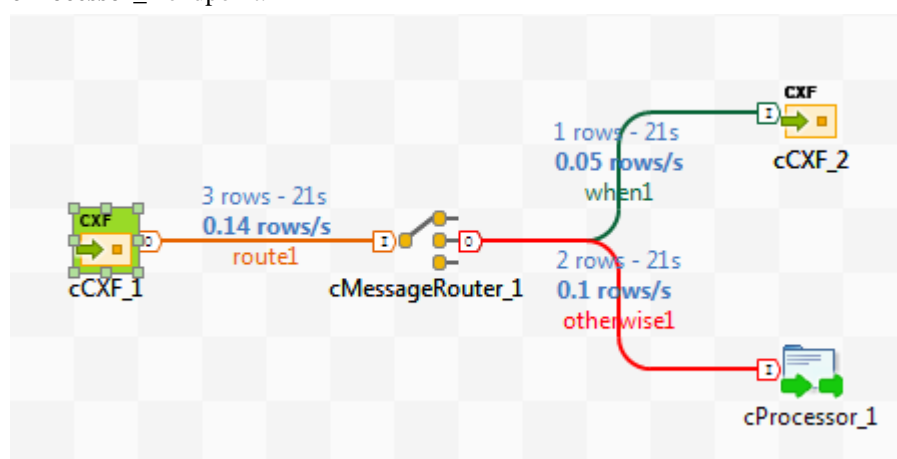
```
[statistics] connecting to socket on port 3815
[statistics] connected
{http://schemas.xmlsoap.org/soap/envelope/}Server|user name error|||
<?xml version="1.0" encoding="UTF-8"?>
<tns:SayHelloServiceOperationResponse
xmlns:tns="http://www.talend.org/service/"><out>Hello
Alice!</out></tns:SayHelloServiceOperationResponse>
{http://schemas.xmlsoap.org/soap/envelope/}Server|user name error|||
[statistics] disconnected
```

☐ Line limit 100 ☒ Wrap

The main grid shows "3 rows", one message being passed along and two going to fault.



4. Now look at the **Route SayHelloRoute 0.1** tab, 1 message went to the **cCXF_2** provider and 2 messages went to the **cProcessor_1** endpoint.



That is the SayHelloRoute demo completed.



Chapter 4. Profiling, cleansing and monitoring data

This chapter aims at users of *Talend Data Quality* who seek a real-life use case to help them take full control over data quality products.

It describes how to use the **Profiling** perspective in *Talend Studio* to profile and cleans data, *Talend Data Quality Portal* to generate a data evolution report and share it with other business users, and finally *Talend Administration Center* to deploy a Job that can launch the data evolution report.

4.1. Profiling customer data

Incorporating appropriate data quality tools in your business processes is vital at the beginning of any project and through the project plan in order to see what type of data quality you have and decide how and what data to resolve.

Suppose, for example, that you want to start a campaign for your sales and marketing groups, or you need to contact customers for billing and payment and your main source to contact appropriate people is email and postal addresses. Having consistent and correct address data is vital in such campaign to be able to reach all people.

This section provides an example of profiling US customer email and postal addresses. It shows how to identify anomalies in address columns, how to use some **Talend** Jobs to recuperate duplicate and non-match addresses and finally how to generate periodic evolution reports to keep monitoring data evolution and share such statistics with business users.

4.1.1. Identifying data anomalies

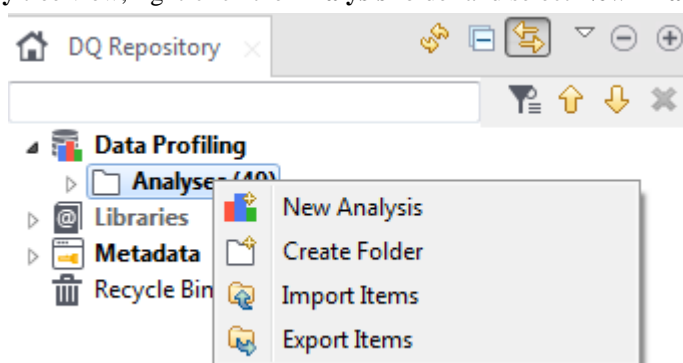
The first step in this example is to profile the customer contact information in a MySQL database. The profiling results provides you with statistics about the values within each column.

4.1.1.1. How to profile address columns

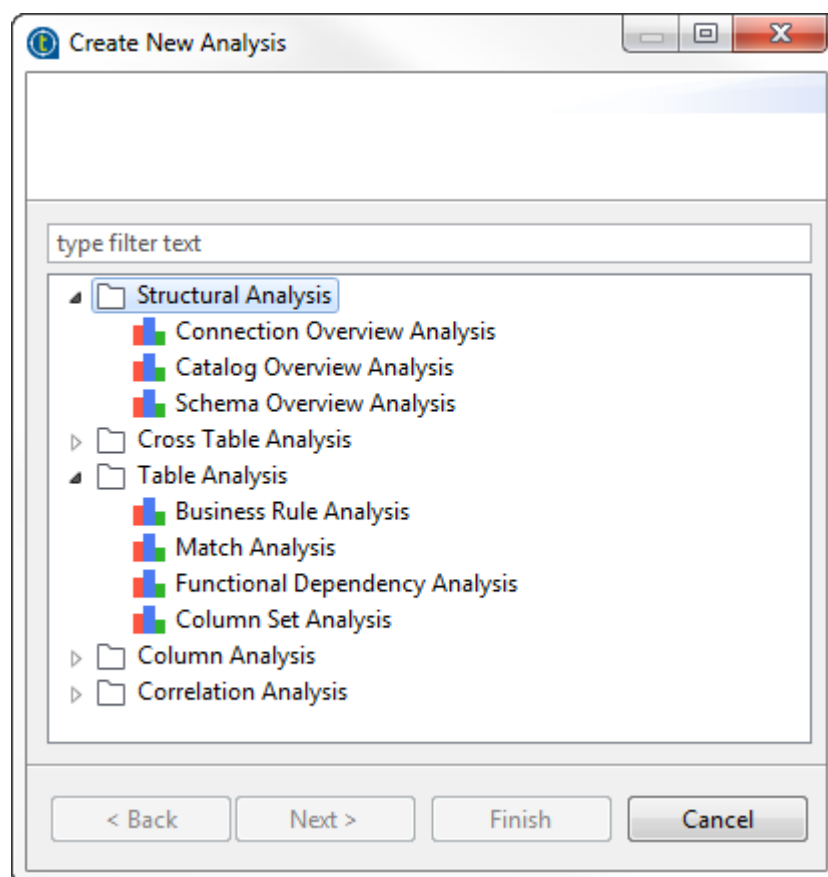
You will use the **Profiling** perspective of the studio to analyze few customer columns including *email* and *postal*. Using out-of-box indicators and patterns on these columns, you can show in the analysis results the matching and non-matching address data, the number of most frequent records for each distinct pattern and the row, duplicate and blank counts in each column.

Defining the column analysis

1. In the **DQ Repository** tree view, right-click the **Analysis** folder and select **New Analysis**.



The **[Create New Analysis]** wizard opens.



2. Start typing *Basic column analysis* in the search field, select **Basic Column Analysis** from the list and click **Next**.

3. In the **Name** field, enter a name for the current column analysis.

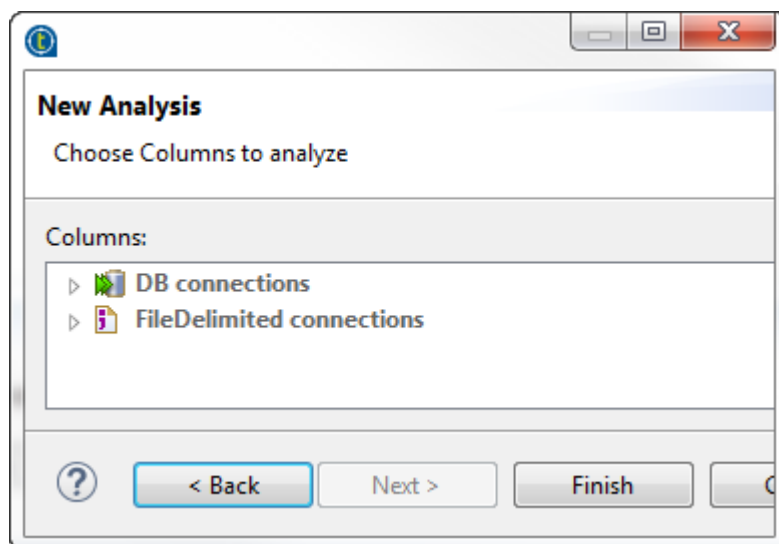


Avoid using special characters in the item names including:

"~", "!", "\", "#", "^", "&", "*", "\\", "/", "?", ":", ";", "\\", ".", "(", ")", "'", "¥", "™", "®", "«", "»", "<", ">".

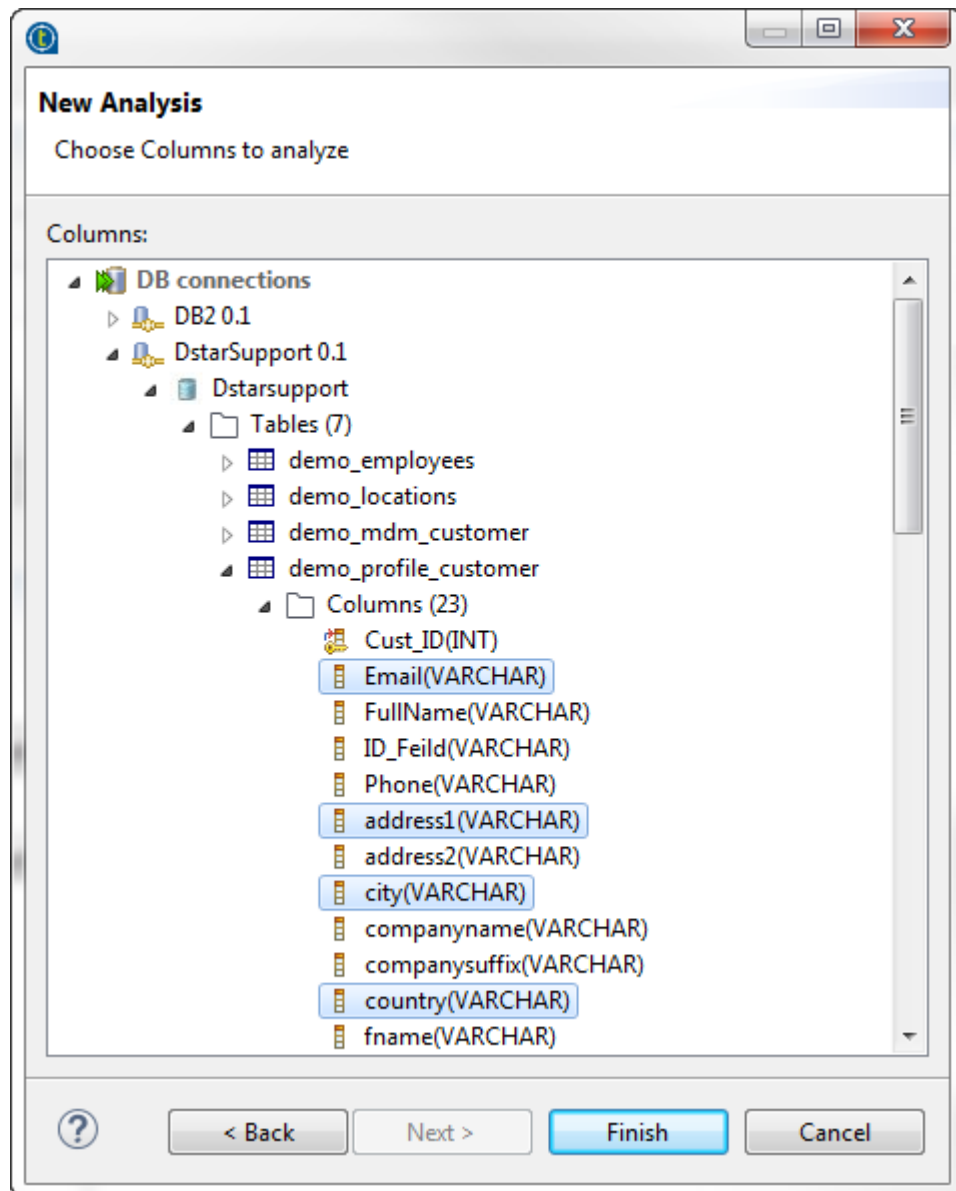
These characters are all replaced with "_" in the file system and you may end up creating duplicate items.

4. Set column analysis metadata (purpose, description and author name) in the corresponding fields and click **Next**.



Selecting the address columns and setting sample data

1. Expand **DB connections** and browse to the address columns you want to analyze.



2. Select the columns and click **Finish** to close the wizard.

A file for the newly created column analysis is listed under the **Analysis** node in the **DQ Repository** tree view, and the analysis editor opens with the analysis metadata.

Column Analysis

▼ **Analysis Metadata**
Set the analysis properties.

Name:

Purpose:

Description:

Author:

Status:

▼ **Data preview**

Connection: Version: 0.1

Limit:

	Email	postal	city	state	country	address1
1	DebraEvans@fa...	16054	Saint Petersburg	PA	US	5870 E EVANS CT
2	TeresaBailey@fa...	94188	San Francisco	CA	US	5411 S THROOP ST
3	JeanMiller@gma...	5477	Richmond	VT	US	8004 E WASHINGTON S
4	HenryMartin@g...	23642	Virginia Beach	VA	US	6383 NW SCHICK PL
5	SandraMorgan@...	26036	Dallas	WV	US	9849 W ST CLAIR ST
6	EvelynWalker@g...	5841	Greensboro	VT	US	8738 S ACADEMY PL
7	KathleenFoster@...	89199	Las Vegas	NV	US	10900 SW BANKS ST
8	BrendaBaker@h...	17501	Akron	PA	US	4420 W CULLERTON ST
9	ShirleyBrown@fr...	23642	Virginia Beach	VA	US	5282 NW WILSON AV

3. In the **Data preview** view, click **Refresh Data**.

The data in the selected columns is displayed in the table.

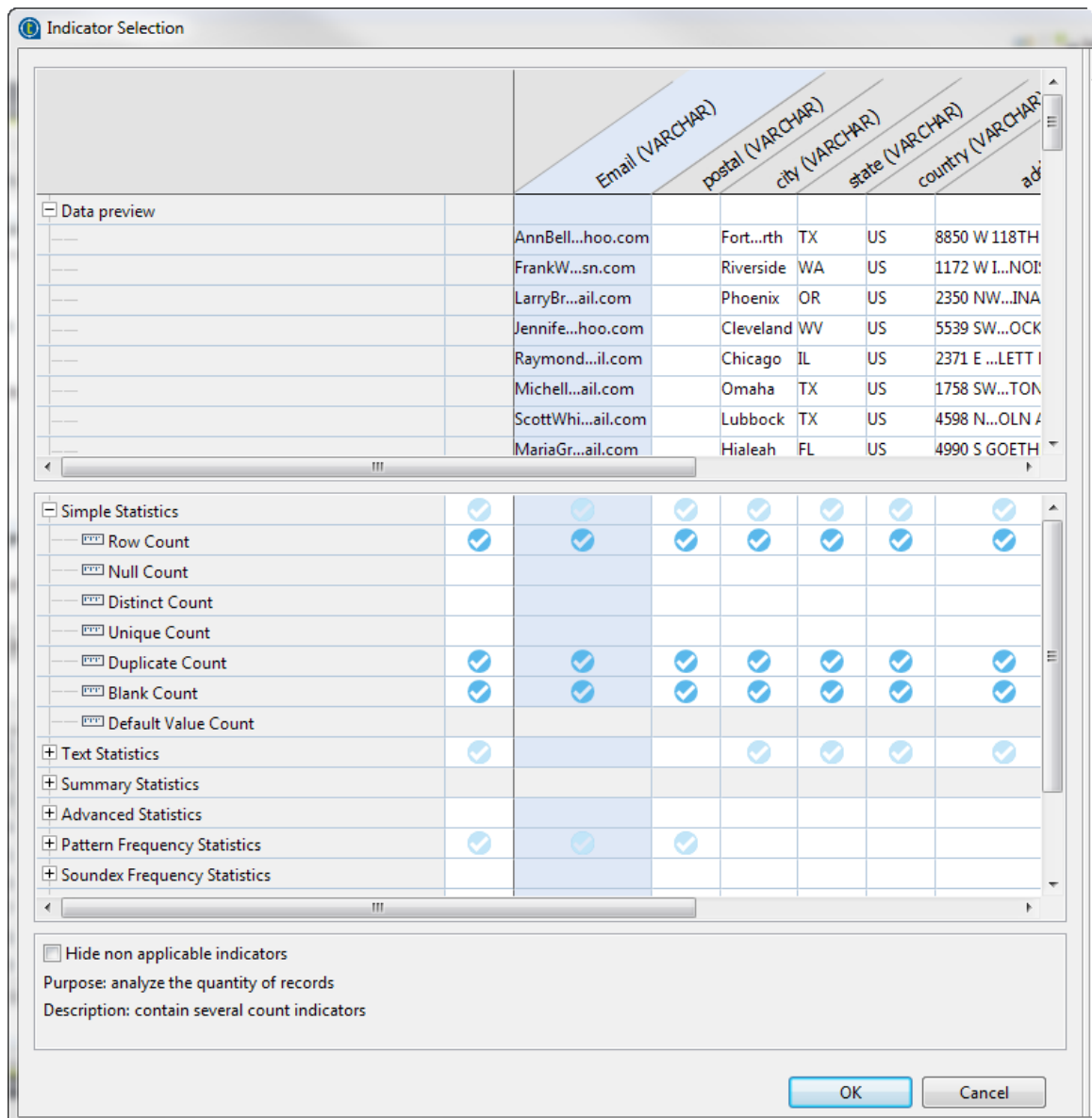
You can change your data source and your selected columns by using the **New Connection** and **Select Data** buttons respectively.

4. In the **Limit** field, set to *50* the number for the data records you want to display in the table and use as sample data.
5. Select **n random rows** to list *50* random records from the selected columns.

For further information on column analysis, see the chapter about column analysis in *Talend Studio User Guide*.

Setting system indicators

1. From the **Data preview** view in the analysis editor, click **Select indicators** to open the **[Indicator Selection]** dialog box.



- Click in the cells next to indicators names to set indicator parameters for the analyzed columns and click **OK**.

You want to see the row, blank and duplicate counts in all columns to see how consistent the data is. Also you want to use the **Pattern Frequency Table** indicator on the *email* and *postal* columns in order to compute the number of most frequent records for each distinct pattern or value.

Indicators are added accordingly to the columns in the **Analyzed Columns** view.

Analyzed Columns				
<div> <div>Go <input type="text"/></div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div> <div>1/2</div>				
Analyzed Columns	Datamining Type	Pattern	UDI	Operation
<div> <div></div> <div>Email (VARCHAR)</div> </div> <div> <div></div> <div>Blank Count</div> </div> <div> <div></div> <div>Duplicate Count</div> </div> <div> <div></div> <div>Pattern Frequency Table</div> </div> <div> <div></div> <div>Row Count</div> </div> <div> <div></div> <div>Email Address</div> </div>	Nominal			
<div> <div></div> <div>postal (VARCHAR)</div> </div>	Nominal			
<div> <div></div> <div>city (VARCHAR)</div> </div>	Nominal			
<div> <div></div> <div>state (VARCHAR)</div> </div>	Nominal			
<div> <div></div> <div>country (VARCHAR)</div> </div>	Nominal			

- Click the option icon  next to the **Blank Count** indicator and set 0 in the **Upper threshold** field.

Defining thresholds on indicators is very helpful as it will write in red the count of the null values in the analysis results.

Indicator

Indicator settings

your input is valid.

Indicator Thresholds

Set the desired indicator thresholds

Lower threshold

Upper threshold 0

Set the desired indicator thresholds in percents

Lower threshold(%)

Upper threshold (%) 0

?


Finish

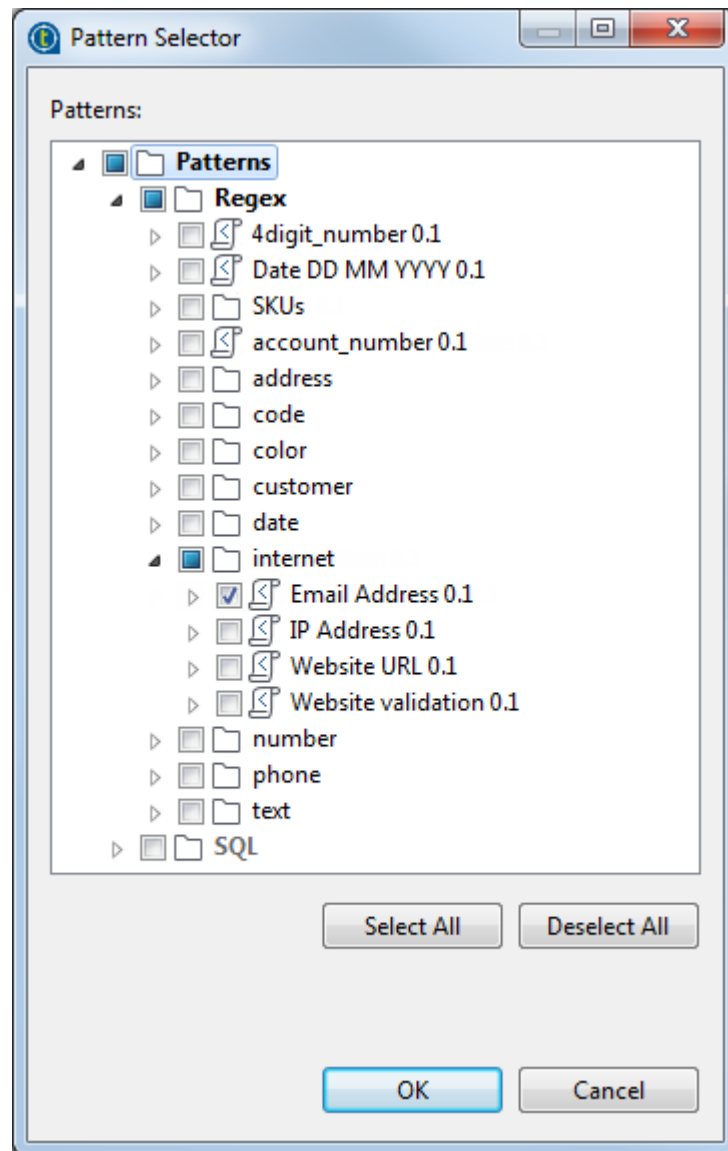
Cancel

For further information on indicator types and their usage when analyzing data, see *Talend Studio User Guide*.


Setting patterns

You would want now to match the content of the *email* column against a standard email format and the *postal* column against a standard US zip code format. This will define the content, structure and quality of emails and zip codes and give a percentage of the data that match the standard formats and the data that does not match.

- In the **Analyzed Columns** view, click the  icon next to *email*.



2. In the **[Pattern Selector]** dialog box, expand **Regex** and browse to **Email Address** in the **internet** folder, and then click **OK**.

3. Click the option icon  next to the **Email Address** indicator and set **98.0** in the **Lower threshold (%)** field.

If the number of the records that match the pattern is fewer than 98%, it will be written in red in the analysis results.

4. Do the same to add to the *postal* column the **US Zipcode Validation** pattern from the **address** folder.

For further information on pattern types and their usage when analyzing data, see *Talend Studio User Guide*.

Executing the analysis and displaying the profiling results

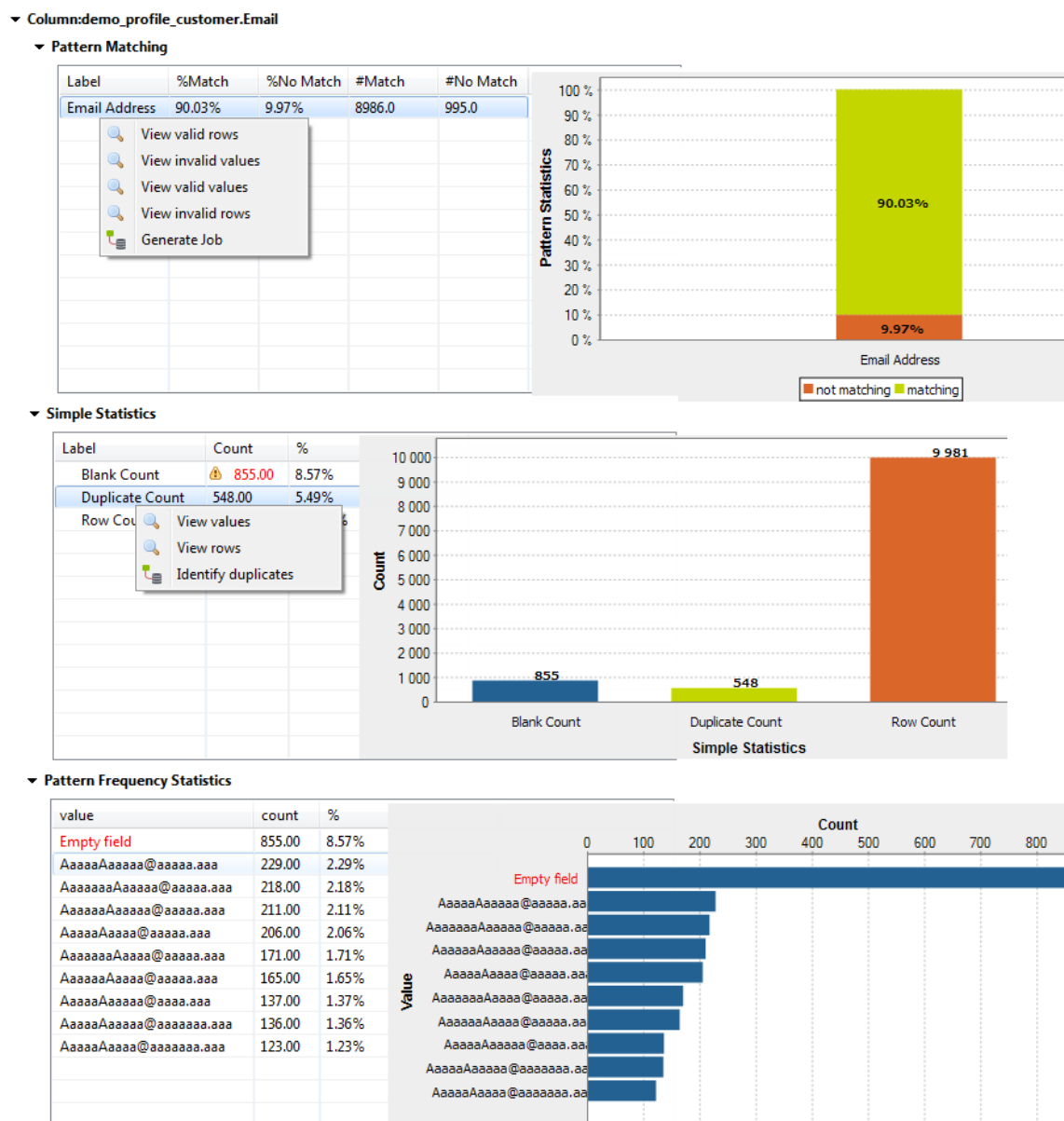
1. Save the column analysis in the analysis editor and then press **F6** to execute it.

A group of graphics is displayed in the **Graphics** panel to the right of the analysis editor showing the results of the column analysis including those for pattern matching.

2. Click the **Analysis Results** tab at the bottom of the analysis editor to access a more detail result view.

These results show the generated graphics for the analyzed columns accompanied with tables that detail the statistic and pattern matching results.

The results for the *email* column look as the following:



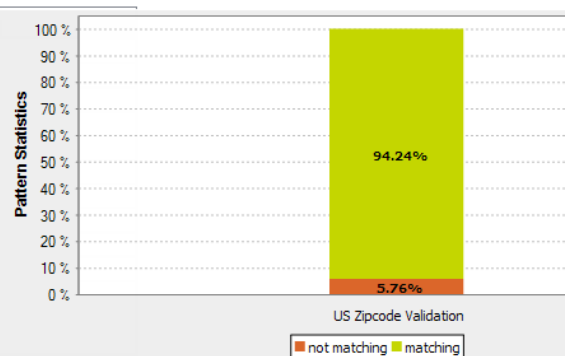
The pattern matching results show that about 10% of the email records do not match the standard email pattern. The simple statistic results show that about 8% of the email records are blank and that about 5% are duplicates. And the pattern frequency results give the number of most frequent records for each distinct pattern. This shows that the data is not consistent and you need to correct and cleans the email data before starting your campaign.

The results for the *postal* column look as the following:

▼ Column:demo_profile_customer.postal

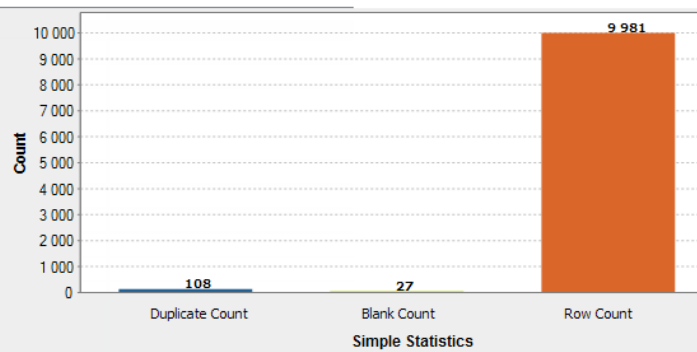
▼ Pattern Matching

Label	%Match	%No Match	#Match	#No Match
US Zipcode Validation	94.24%	5.76%	9406.0	575.0



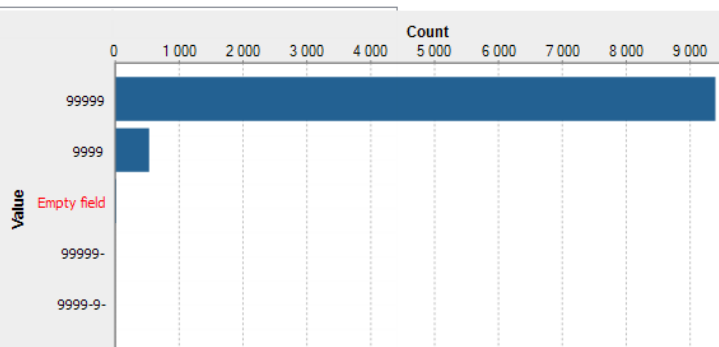
▼ Simple Statistics

Label	Count	%
Duplicate Count	108.00	1.08%
Blank Count	27.00	0.27%
Row Count	9981.00	100.00%



▼ Pattern Frequency Statistics

value	count	%
99999	9406.00	94.24%
9999	540.00	5.41%
Empty field	27.00	0.27%
99999-	7.00	0.07%
9999-9-	1.00	0.01%



The result sets for the *postal* column give the count of the records that match and those that do not match a standard US zip code format. The results sets also give the blank and duplicate counts and the number of most frequent records for each distinct pattern. These results show that the data is not very consistent.

Then some percentage of the customers can not be contacted by either email or US mail service. These results show clearly that your data is not very consistent and that it needs to be corrected.

4.1.1.2. How to view analyzed data

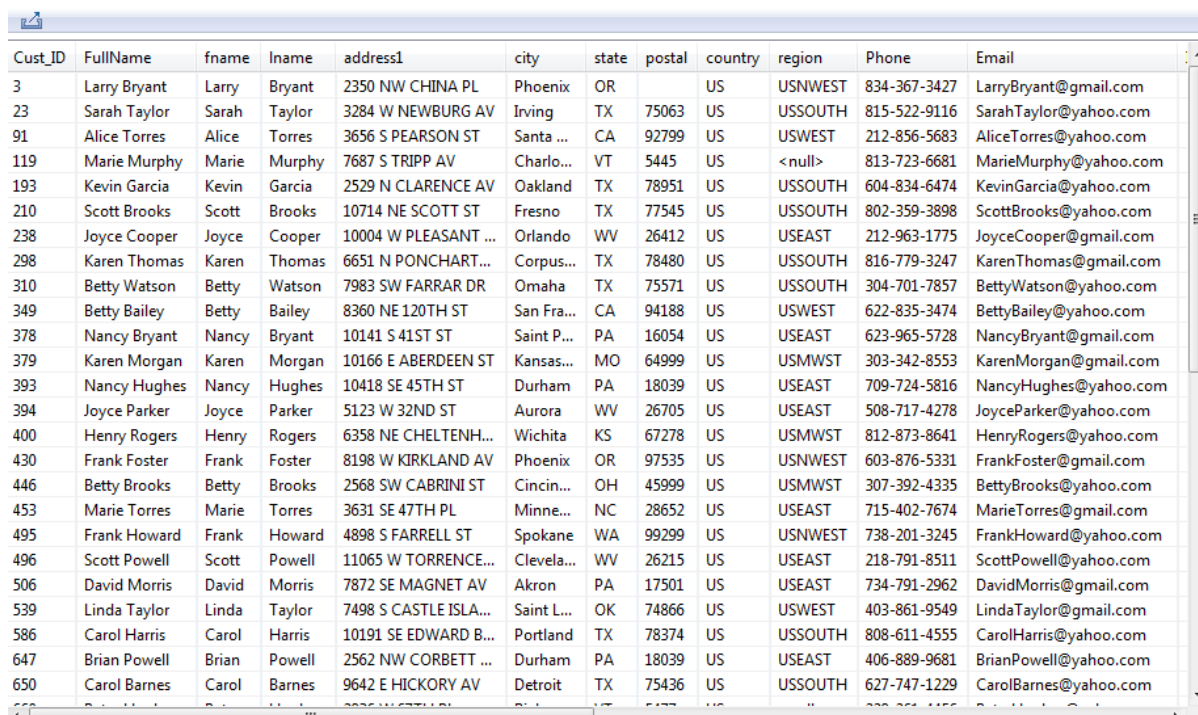
After running the column analysis using the SQL engine and from the **Analysis Results** view of the analysis editor, you can right-click any of the rows/bars in the result tables/charts and access a view of the actual analyzed data. This could be very helpful to see invalid rows for example and start analyzing what needs to be done to clean such data.

To view and export the analyzed data, do the following:

1. At the bottom of the analysis editor, click the **Analysis Results** tab to open a detailed view of the analysis results.

- Right-click the data row in the statistic results of the *Email* column and select **View rows** for example.

The **Data Explorer** perspective opens listing the invalid rows in the *email* column.



Cust_ID	FullName	fname	lname	address1	city	state	postal	country	region	Phone	Email
3	Larry Bryant	Larry	Bryant	2350 NW CHINA PL	Phoenix	OR		US	USNWEST	834-367-3427	LarryBryant@gmail.com
23	Sarah Taylor	Sarah	Taylor	3284 W NEWBURG AV	Irving	TX	75063	US	USSOUTH	815-522-9116	SarahTaylor@yahoo.com
91	Alice Torres	Alice	Torres	3656 S PEARSON ST	Santa ...	CA	92799	US	USWEST	212-856-5683	AliceTorres@yahoo.com
119	Marie Murphy	Marie	Murphy	7687 S TRIPP AV	Charlo...	VT	5445	US	<null>	813-723-6681	MarieMurphy@yahoo.com
193	Kevin Garcia	Kevin	Garcia	2529 N CLARENCE AV	Oakland	TX	78951	US	USSOUTH	604-834-6474	KevinGarcia@yahoo.com
210	Scott Brooks	Scott	Brooks	10714 NE SCOTT ST	Fresno	TX	77545	US	USSOUTH	802-359-3898	ScottBrooks@yahoo.com
238	Joyce Cooper	Joyce	Cooper	10004 W PLEASANT ...	Orlando	WV	26412	US	USEAST	212-963-1775	JoyceCooper@gmail.com
298	Karen Thomas	Karen	Thomas	6651 N PONCHART...	Corpus...	TX	78480	US	USSOUTH	816-779-3247	KarenThomas@gmail.com
310	Betty Watson	Betty	Watson	7983 SW FARRAR DR	Omaha	TX	75571	US	USSOUTH	304-701-7857	BettyWatson@yahoo.com
349	Betty Bailey	Betty	Bailey	8360 NE 120TH ST	San Fra...	CA	94188	US	USWEST	622-835-3474	BettyBailey@yahoo.com
378	Nancy Bryant	Nancy	Bryant	10141 S 41ST ST	Saint P...	PA	16054	US	USEAST	623-965-5728	NancyBryant@gmail.com
379	Karen Morgan	Karen	Morgan	10166 E ABERDEEN ST	Kansas...	MO	64999	US	USMWST	303-342-8553	KarenMorgan@gmail.com
393	Nancy Hughes	Nancy	Hughes	10418 SE 45TH ST	Durham	PA	18039	US	USEAST	709-724-5816	NancyHughes@yahoo.com
394	Joyce Parker	Joyce	Parker	5123 W 32ND ST	Aurora	WV	26705	US	USEAST	508-717-4278	JoyceParker@yahoo.com
400	Henry Rogers	Henry	Rogers	6358 NE CHELTENH...	Wichita	KS	67278	US	USMWST	812-873-8641	HenryRogers@yahoo.com
430	Frank Foster	Frank	Foster	8198 W KIRKLAND AV	Phoenix	OR	97535	US	USNWEST	603-876-5331	FrankFoster@gmail.com
446	Betty Brooks	Betty	Brooks	2568 SW CABRINI ST	Cincin...	OH	45999	US	USMWST	307-392-4335	BettyBrooks@yahoo.com
453	Marie Torres	Marie	Torres	3631 SE 47TH PL	Minne...	NC	28652	US	USEAST	715-402-7674	MarieTorres@gmail.com
495	Frank Howard	Frank	Howard	4898 S FARRELL ST	Spokane	WA	99299	US	USNWEST	738-201-3245	FrankHoward@yahoo.com
496	Scott Powell	Scott	Powell	11065 W TORRENCE...	Cleavela...	WV	26215	US	USEAST	218-791-8511	ScottPowell@yahoo.com
506	David Morris	David	Morris	7872 SE MAGNET AV	Akron	PA	17501	US	USEAST	734-791-2962	DavidMorris@gmail.com
539	Linda Taylor	Linda	Taylor	7498 S CASTLE ISLA...	Saint L...	OK	74866	US	USWEST	403-861-9549	LindaTaylor@gmail.com
586	Carol Harris	Carol	Harris	10191 SE EDWARD B...	Portland	TX	78374	US	USSOUTH	808-611-4555	CarolHarris@yahoo.com
647	Brian Powell	Brian	Powell	2562 NW CORBETT ...	Durham	PA	18039	US	USEAST	406-889-9681	BrianPowell@yahoo.com
650	Carol Barnes	Carol	Barnes	9642 E HICKORY AV	Detroit	TX	75436	US	USSOUTH	627-747-1229	CarolBarnes@yahoo.com

4.1.2. Sharing analysis results: reports

After profiling the email and zip code columns and getting the detail results about the structure and consistency of the address data, you need to share these results with other business users.

You must first generate a report file on the analysis results from the studio and save the report in a data quality data mart. Business users can then access the report from *Talend Data Quality Portal*, which is a web-based platform that shares analysis results generated from the studio and saved in the data quality data mart.

In the **Profiling** perspective of the studio:

- In the **DQ Repository** tree view, right-click the analysis name and select **New Report**.

The report editor is displayed with the selected analysis listed in the **Analysis List**.

Report Settings

▼ **Report Metadata**
Set the properties of the report.

Name:

Purpose:

Description:

Author:

Status:

▼ **Analysis List**
[Select analyses](#)

Analysis	Execution Date	Refresh	Template type	Remove
profile_customer		<input checked="" type="checkbox"/>	Evolution <input type="text" value=""/>	<input type="button" value="Browse..."/> <input type="button" value="Remove"/>

☒ Refresh All

► **Generated Report Settings**

► **Presentation Settings**

► **Database Connection Settings**

Report Settings

- In the **Analysis list** view and from the **Template type** list, select **Evolution** as the type for the report you want to generate.

In this example, you want to generate an evolution report which provides information showing the evolution through time of the indicators used on the *email* and *postal* columns. This report allows you to compare current and historical statistics to determine the improvement or degradation of the address data. Such information is vital to decide to intervene and resolve data at the right time and thus monitor the quality of data on an on-going basis.

- Select the **Refresh All** check box to refresh the listed analysis before generating the report.
- In the **Generated Report Settings** view and from the **File Type** list, select to generate a pdf report file.
- In the **Database Connection Settings** view, set the connection parameters to the data mart where you want to store the report results.

▼ **Database Connection Settings**

Db Type:

Db Version:

Host:

Port:

Db Name:

User:

Password:


Url:

Driver:

Dialect:

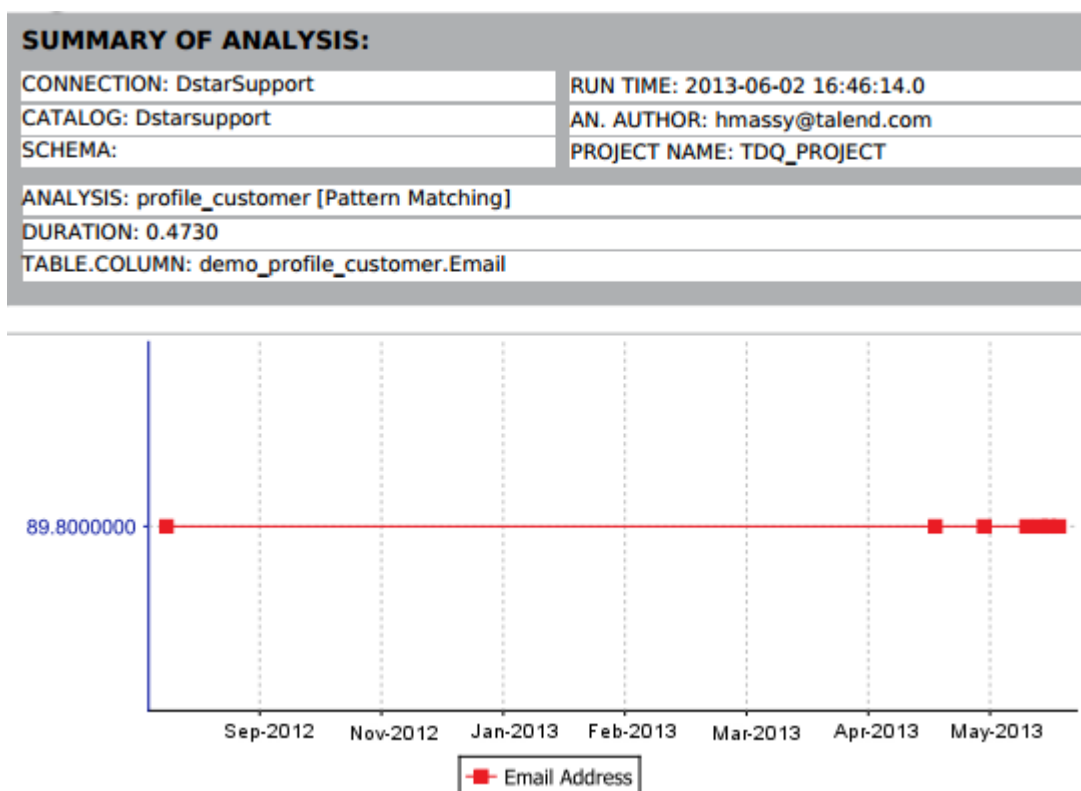
- Click the **Check** button to verify if your connection is successful.

A message confirms if the database exists and if the connection is successful.

- If the database structure does not exist, click **OK** in the message to let the studio creates it for you.
- Click **OK** to close the confirmation message.
- Save the report and click  on the editor toolbar to generate the report file.

A report file is generate and listed under the **Reports** node in the **DQ Repository** tree view. The report shows the evolution through time of the simple statistics indicators and the patterns used on the *email* and *postal* columns.

Below are the results of the *email* column:



This chart shows that 89.80% of the email addresses are valid right now.

SUMMARY OF ANALYSIS:

CONNECTION: DstarSupport

RUN TIME: 2013-06-02 16:46:14.0

CATALOG: Dstarsupport

AN. AUTHOR: hmassy@talend.com

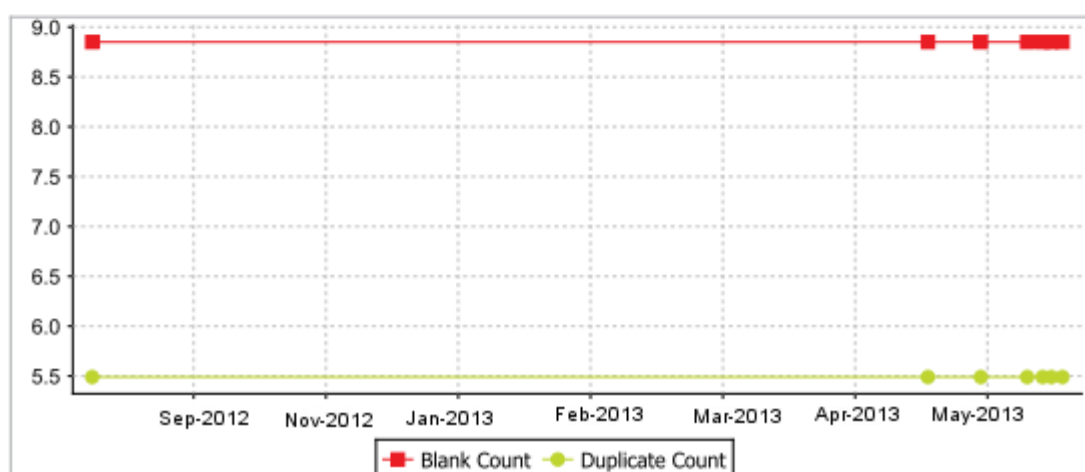
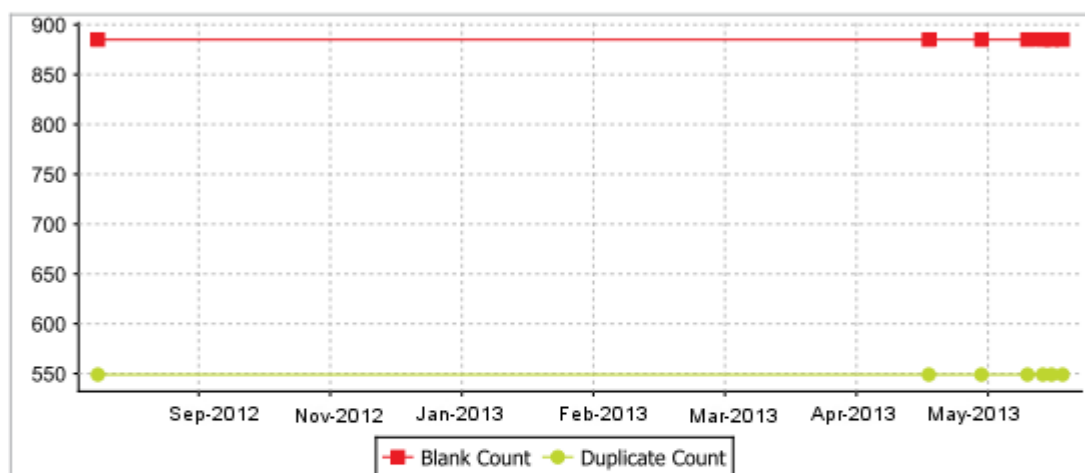
SCHEMA:

PROJECT NAME: TDQ_PROJECT

ANALYSIS: profile_customer [Simple Statistics]

DURATION: 0.4730

TABLE.COLUMN: demo_profile_customer.Email



For the simple statistics indicators, there are two charts: the first indicates the change in the statistics and the second indicates the percentage of that change.

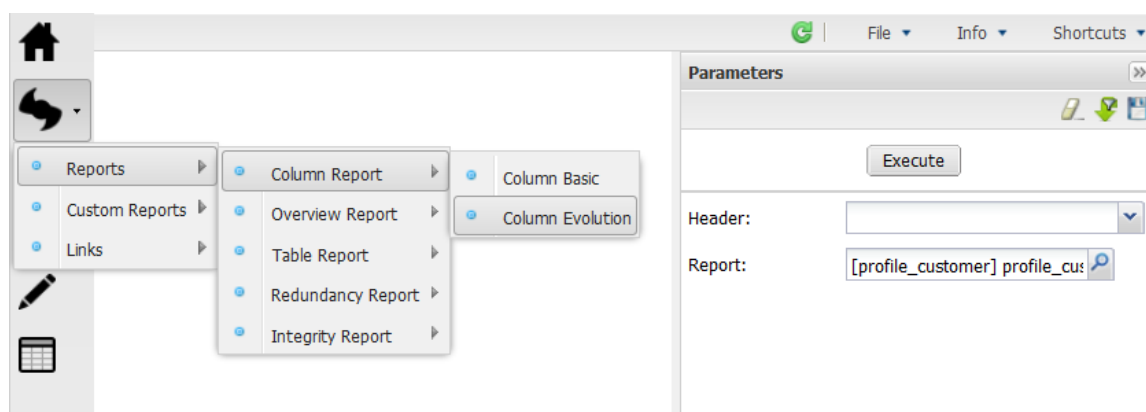
Generating this report repeatedly will give a flat line if there is no change in data. The line will start to go upwards if data is fixed and downwards if data gets less accurate and consistent.

For further information on reports, see the Reports chapter in *Talend Studio User Guide*.

After generating this report in the studio, business users can access it from *Talend Data Quality Portal*.

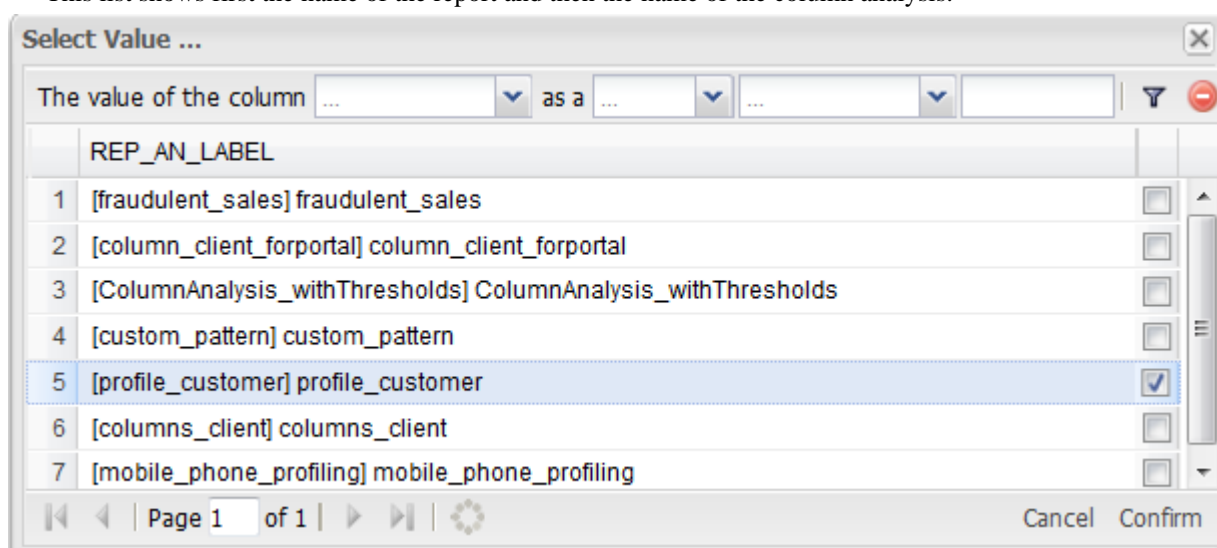
To generate the evolution report from the Portal:

1. Access *Talend Data Quality Portal* using `tdq_user` as username and `tdq` as password.



2. Click the **User menu** and slide the cursor on **Reports > Column Report > Column Evolution**.
3. Click the **Report** explore icon.

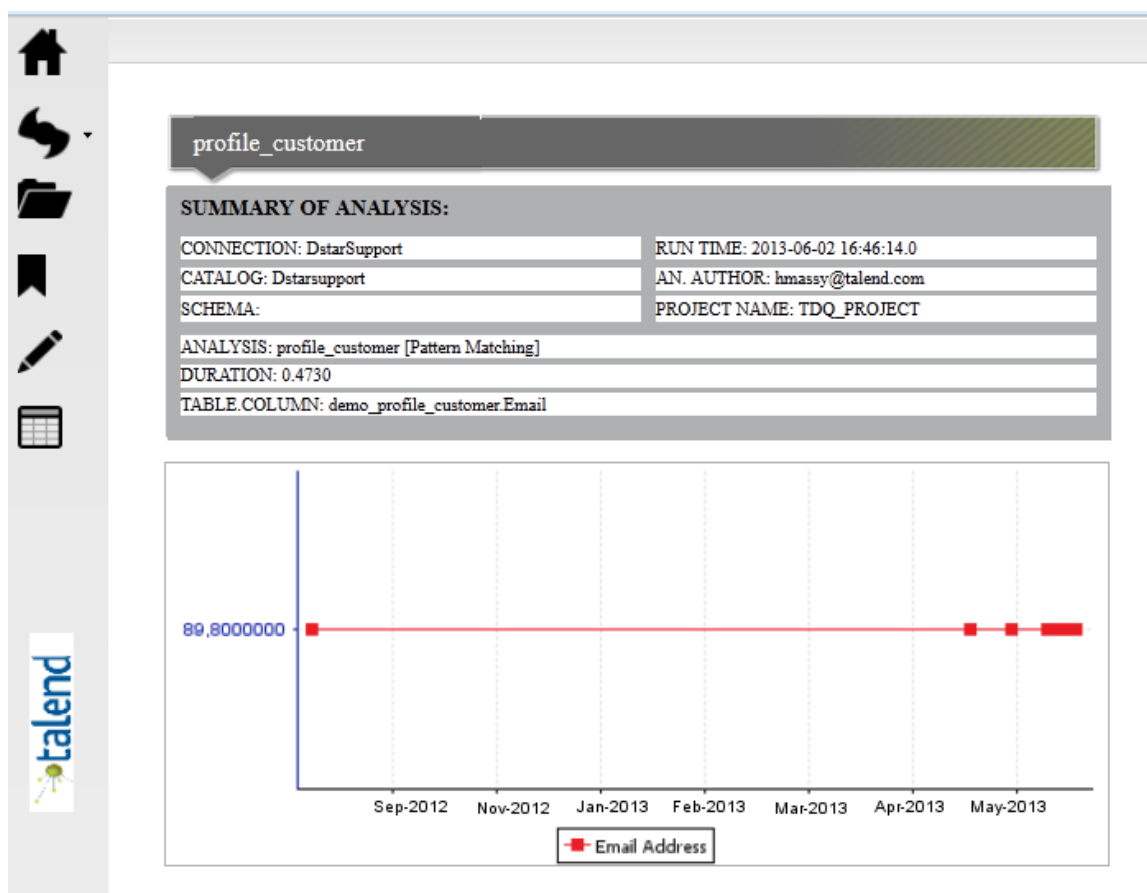
A dialog box opens to list all evolution reports generated on column analyses in the *Profiling* perspective. This list shows first the name of the report and then the name of the column analysis.



4. Select the check box of the evolution report you want to generate and then click **Confirm** at the bottom right corner of the dialog box.
5. Click **Execute** at the top of the **Parameters** panel.

A loading indicator is displayed and then the report is open in the page.

You will have in the Portal the same profiling results you generated from the studio:



For further information on the Portal, see the *Talend Data Quality Portal User and Administrator Guide*.

4.2. Cleansing data

After profiling customer data and identifying its problems, some actions should be taken on data to cleans it. You may start by generating two **Talend Jobs**: one to remove duplicates from the *email* column and the other to remove the values that do not match the email pattern.

This will help you seeing what to resolve and then you can decide what tool to use to intervene and resolve these address issues.

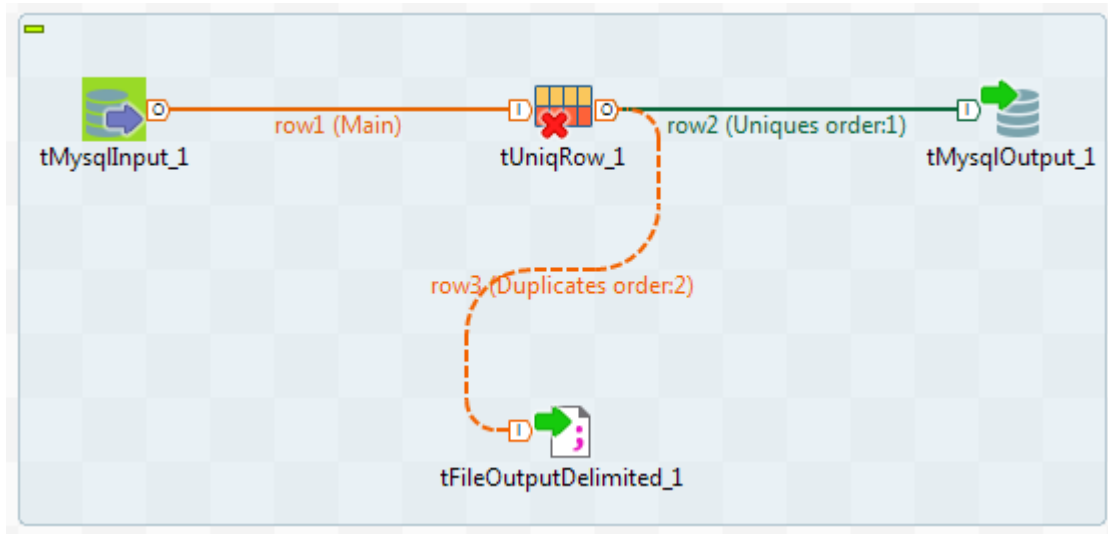
4.2.1. Removing duplicate values

After analyzing the *email* and *postal* columns using simple statistics indicators, the analysis results show the number of duplicate records in the columns. You can generate a ready-to-use Job on the analysis results. This Job removes duplicate values in the selected column.

To remove duplicate values from the *email* column:

1. In the **Profiling** perspective, click **Analysis Results** at the bottom of the editor.
2. In the **Simple Statistics** results of the *email* column, right-click the duplicate count bar in the chart and select **Remove duplicates**.

The **Integration** perspective opens in the studio showing the generated Job with the corresponding components. For more information on such components, see *Talend Components Reference Guide*.



The database input component and the **tUniqueRow** components are already configured according to your connection and the columns you are analyzing.

3. Save the Job and press **F6** to execute it.

Duplicate values are written to the specified output database and file.

You can follow the same procedure to remove duplicates from the *postal* column.

For further information on using the **Profiling** perspective to identify and remove corrupt, incomplete or inaccurate data, see the chapter about data cleansing in *Talend Studio User Guide*.

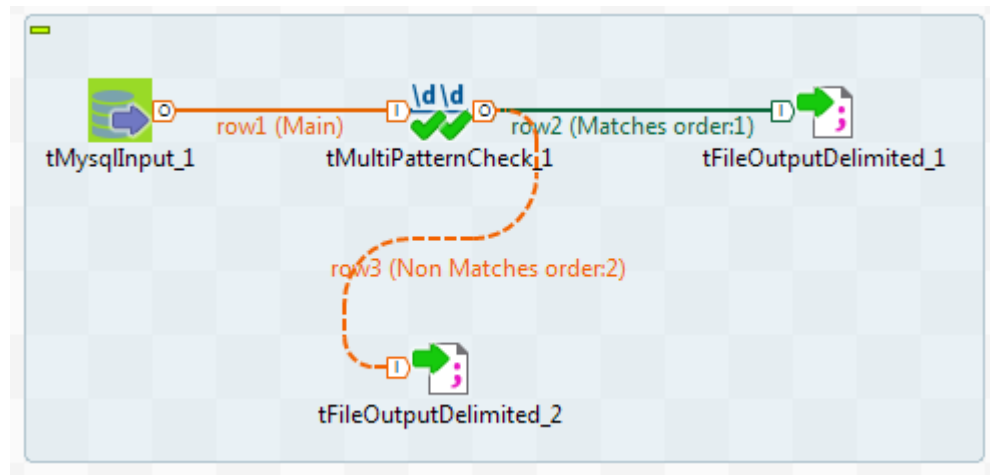
4.2.2. Removing non-matching values

The email pattern used on the *email* column showed that some records do not respect the standard email format. You can generate a ready-to-use Job to recuperate the non-matching rows from the column.

To recuperate non-matching email rows:

1. In the **Profiling** perspective, click the **Analysis Results** tab at the bottom of the editor.
2. In the **Pattern Matching** results of the *email* column, right-click the chart bar or the numerical results and select **Generate Job**.
3. In the open dialog box, select **Generate an ETL Job to handle rows**.

The **Integration** perspective opens on the generated Job.



This Job uses the Extract Transform Load process to write in two separate output files the valid/invalid email rows that match/do not match the pattern.

4. Save the Job and press **F6** to execute it.

The valid and invalid rows of the *email* column are written in the defined output files.

You can replace the output files with different **Talend** components and recuperate the valid/invalid email rows and write them in databases for example.

You can follow the same procedure to recuperate invalid rows from the *postal* column.


For further information on using the **Profiling** perspective to identify and remove corrupt, incomplete or inaccurate data, see the chapter about data cleansing in *Talend Studio User Guide*.



Appendix A. Glossary

When working with *Talend Studio* and in order to understand its functional mechanism, it is important to understand some basic vocabulary.

component	<p>A component is an executable part of a Job or Route used to connect to an external source or perform a specific data integration operation, no matter what data sources you are integrating: databases, applications, flat files, Web services, etc. A component can minimize the amount of hand-coding required to work on data from multiple, heterogeneous sources.</p> <p>Components are grouped in families according to their usage and displayed in the Palette of the Integration perspective of <i>Talend Studio</i>.</p> <p>For detailed information about components types and what they can be used for, see <i>Talend Components Reference Guide</i>.</p>
item	<p>An item is the fundamental technical unit in a project. Items are grouped, according to their types, as: Job Design, Business model, Context, Code, Metadata, etc. One item can include other items. For example, the business models and the Jobs you design are items, metadata and routines you use inside your Jobs are items as well.</p>
Job	<p>A Job is a graphical design, of one or more components connected together, that allows you to set up and run dataflow management processes. It translates business needs into code, routines and programs. Jobs address all of the different sources and targets that you need for data integration processes and all other related processes.</p>
Joblet	<p>A Joblet is a specific component that replaces Job component groups. It factorizes recurrent processing or complex transformation steps to ease the reading of a complex Job. Joblets can be reused in different Jobs or several times in the same Job.</p>
metadata	<p>Metadata is information that describes the characteristics of any data object, such as its name, type, location, author, date created, size, and so on, together with relationships with other data objects that the enterprise has to manage or that an IT tool may generate. Metadata can be created manually or automatically by a system.</p>
project	<p>Projects are structured collections of items and their associated metadata. All of the Jobs and business models you design are organized in Projects.</p>
repository	<p>A repository is the storage location <i>Talend Studio</i> uses to gather data related to all of the technical items that you use either to describe business models or to design Jobs.</p> <p><i>Talend Studio</i> can connect to as many local or remote repositories as needed.</p>

Route	A Camel Route is a graphical design, based on Apache Camel framework, of two or more components connected together that allows you to set up and run routing and mediation rules. A routing rule defines how messages will be moved from one service (or endpoint) to another.
Service	A Service is a graphical design, of several WSDL objects (service, binding, port type and so on) linked together, that allows you to set up and implement Web services. A Service is associated with one or more data service Jobs as the service provider and can be consumed by consumer Jobs.
service Job	<p>A data service Job is a graphical design, of one or more components connected together, that allows you to set up and run data service processes. It translates business needs into code, routines and programs. Jobs address all of the different sources and targets that you need for data integration processes and combine it with Web services.</p> <p> Data service Jobs will simply be referred to as Jobs in the following documentation.</p>
workspace	A workspace is the directory where you store all your project folders. You need to have one workspace directory per connection (repository connection). <i>Talend Studio</i> enables you to connect to different workspace directories, if you do not want to use the default one.