ProjectPro

Talend Real-Time Project for ETL Process Automation

CookBook

Table of Content:

Introduction

In this training, we will discuss a file-loading automation process using an ETL tool called Talend. The case study for the proposed pipeline can be as follows:

Now imagine a scenario where you receive 10 files in a folder daily. And someone is sitting manually processing those files every day. This could be files like surveys, sales, or financial data from any bank. The process would be the same for all these files. The driver load only takes care of the loading and transformation process, but it will eliminate manual intervention. It will reduce processing time, and it'll also eliminate all processing errors. Also, you don't have to wait for files to be posted on a folder or FTB. This job can schedule at multiple intervals, and whenever a file is posted, it will process it.

File automation is needed in various scenarios to streamline manual processes and increase efficiency, reduce errors, and save time. Some of the key benefits of file automation include:

1. **Improved accuracy:** Automated file processing reduces the risk of errors that can occur during manual processing, such as typos or incorrect data entry.
2. **Increased speed:** Automated file processing can handle large volumes of data much faster than manual processing, reducing the time it takes to complete a task.
3. **Improved efficiency:** Automated file processing eliminates the need for manual data entry, freeing up staff time to focus on other important tasks.
4. **Better data management:** Automated file processing ensures that data is consistently organized and processed in a specific way, improving data quality and making it easier to find and analyse data.
5. **Reduced costs:** Automated file processing eliminates the need for manual labour, reducing labour costs and increasing productivity.
6. **Increased scalability:** As businesses grow, the volume of data they need to process can become overwhelming. Automated file processing enables businesses to scale up and handle increasing amounts of data with ease.

In short, file automation provides a more efficient and effective way to handle large volumes of data and enables businesses to keep pace with growing data demands.

Before we jump on the job, let's cover the technology stack for the ETL pipeline; we will use Talent.

Talend Open Studio is a free, open-source data integration tool for extracting, transforming, and loading (ETL) data into various data stores. It provides a visual interface and a wide range of pre-built connectors and components for connecting to various data sources, such as databases, applications, and cloud services, and performing data transformations. The tool supports several big data technologies, including Apache Hadoop and Apache Spark, making it an ideal solution for processing big data. The data integration processes can be automated, making it easy to schedule data extraction, transformation, and loading. Talend Open Studio is a highly scalable and reliable tool, providing a comprehensive set of features for data integration and big data processing, while its open-source nature allows for customization and integration with other tools. The tool also supports data quality and data governance capabilities, ensuring the accuracy and completeness of data throughout the data integration process.

With its ease of use, scalability, and powerful data processing capabilities, Talend Open Studio is a popular choice for businesses looking for a cost-effective and flexible data integration solution.

The prerequisites for Talend Open Studio are as follows:

1. **Operating System:** Talend Open Studio is compatible with Windows, Linux, and macOS operating systems. It is recommended to use a 64-bit operating system for improved performance.
2. **Java:** Talend Open Studio requires Java 8 or higher to run. It is recommended to install the latest version of Java for improved performance and stability.
3. **Hardware:** Talend Open Studio requires a minimum of 2 GB of RAM and 2 GB of free hard disk space. However, the amount of RAM and hard disk space required may vary depending on the size and complexity of the data processing tasks being performed.
4. **Database Connectivity:** To connect to a database, you will need to have the appropriate JDBC driver installed. Talend Open Studio supports various databases, including Oracle, MySQL, SQL Server, PostgreSQL, and more.
5. **Network Connectivity:** Talend Open Studio requires network connectivity to access remote data sources and databases. Ensure that you have the necessary permissions to access these resources.
6. **Optional Components:** Talend Open Studio can be integrated with various other tools and systems, such as Apache Hadoop and Apache Spark. If you plan to use these tools, you will need to install and configure them as well.

It is also important to note that Talend Open Studio requires a basic understanding of data integration concepts, SQL, and Java programming. You can access the Talend community and guidance documentation if you are new to data integration or need additional support.

The Talend Open Studio interface consists of several key components:

1. **Repository:** This is where all the data integration jobs, metadata, and context variables are stored.
2. **Palette:** This is a collection of pre-built components and connectors that can be used to build data integration jobs. The palette is organized by categories, such as database, file, and big data components.
3. **Job Designer:** This is where you can design and build data integration jobs using the components from the Palette. The Job Designer provides a visual interface for connecting components, defining data transformations, and setting up data flows.
4. **Contexts:** This is where you can define and manage context variables, which are used to dynamically control a job's behaviour at runtime.
5. **Run View:** This is where you can execute, monitor, and manage your data integration jobs. The Run View provides real-time information about the status of your jobs, as well as detailed logs and statistics.
6. **Metadata:** This is where you can manage the metadata associated with your data integration jobs, such as database connections, tables, columns, and schemas.
7. **Outline View:** This is a hierarchical view of the components and connections in your data integration job, allowing you to navigate and manage your job quickly.

The Talend Open Studio interface is highly customizable and can be configured to meet your specific needs. Additionally, it provides a range of features for managing and testing your data integration jobs, including version control, debugging tools, and reporting capabilities.

Did you know?

Talend is a data integration and management software company that provides solutions for big data, cloud storage, data management, and data quality. It was founded in 2005 and has its headquarters in Redwood City, California. One interesting fact about Talend is that it has a low-code platform for data integration and management, making it easier for users with limited technical skills to access and manage data.

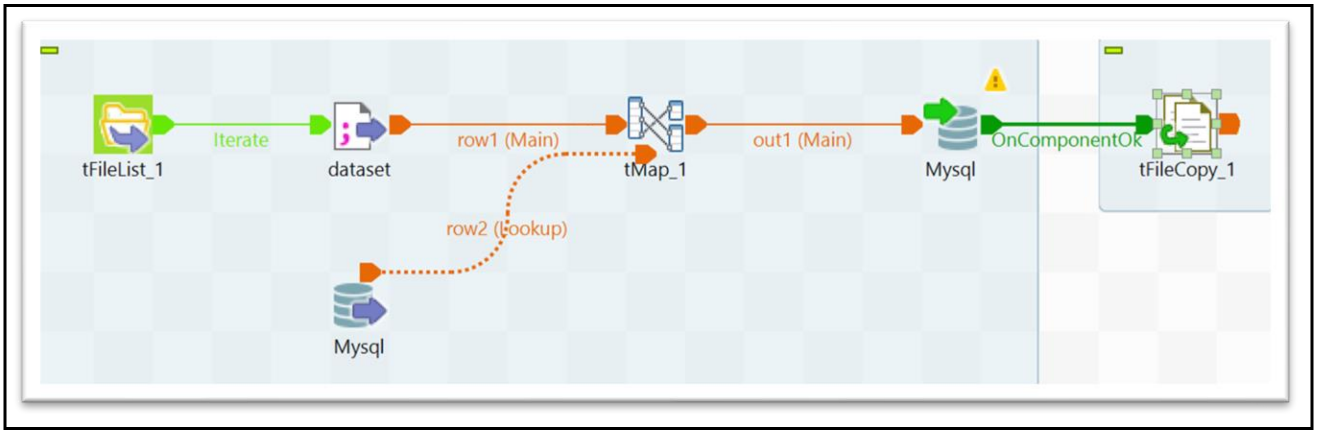
Use-case depicted in this project

In this project, we will create an ETL pipeline in Talend Open Studio to automate the process of File Loading and Processing. We will also schedule the task using Task Scheduler to trigger the event after a certain period. This project will help you learn all the basics of Talend.

This Project uses the Sales order dataset, which includes a few of the following fields:

- Order_Date
- Brand
- Sneaker
- Sale
- Retail
- Profit
- ProfitPercent
- Buyer_Region

Following is the Talend job that we will use in this project:



There are six component components used in this job, so let's go through each one of them.

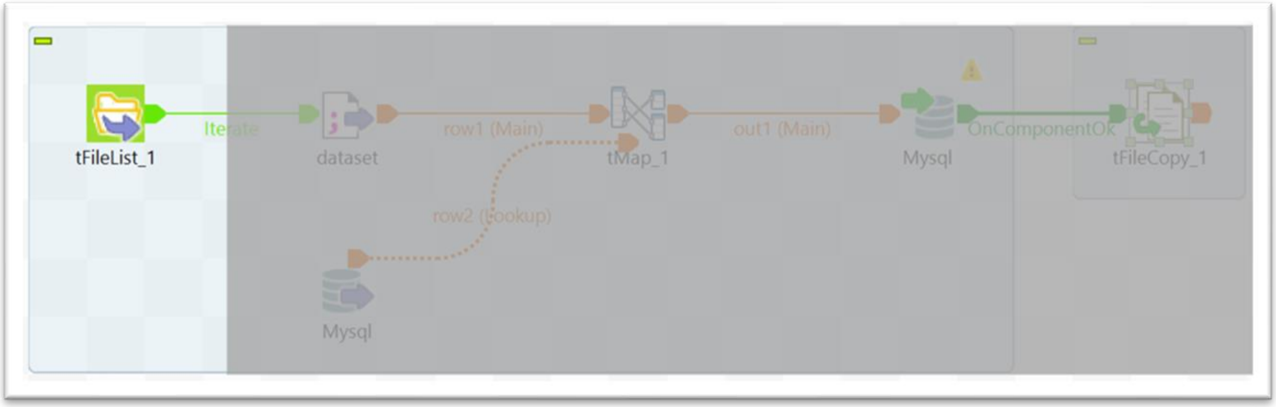
1. The first component is the file reader(**tFileList_1**), which basically reads all the files in the folder. So, it works as a loop.
2. The second component is the tFileInputDelimited(**dataset**). The tFileInputDelimited is a Talend component that reads data from delimited text files, such as CSV files. So, this component reads the files iteratively coming from tFileList_1.
3. The third component is the **tDBInput(MySQL)**, which is used to read data from a MySQL database. There's a table stored in MySQL database named state_map, which contains 2 columns i.e., state and abbreviation. For example, a row of the table contains 'Alabama', which is a state, and 'AL' which is its abbreviation. The tMysqlInput component read data from this table and send it to next component.
4. The next component is the mapping component(**tMap_1**), which is used for transformation. The tMap component takes as input one or more flows of data records. It generates an output flow of data records based on the transformations and manipulations specified in the tMap

configuration. We will merge the data from tMysqlInput and dataset components using the abbreviations of the state and insert the state abbreviation in our final table.

5. The last components are **tDBOutput(MySQL)** and **tFileCopy**.
   The tDBOutput(MySQL) component in Talend Open Studio is used to write data to a MySQL database. With the tMysqlOutput component, you can specify the database connection details, the target table, and the data to be written to the database. The tMysqlOutput component loads the final table into the database.
   The tFileCopy component in Talend Open Studio is used to copy files from one location to another. The tFileCopy component inputs the source and target files and performs a file copy operation from the source to the target.
   Basically, once the file is processed and loaded, we don't want it to stay in the same folder. We want it to be moved into another folder. So next time, when our job runs, it only processes the new files.

Now, let’s deep dive into each component.

tFileList



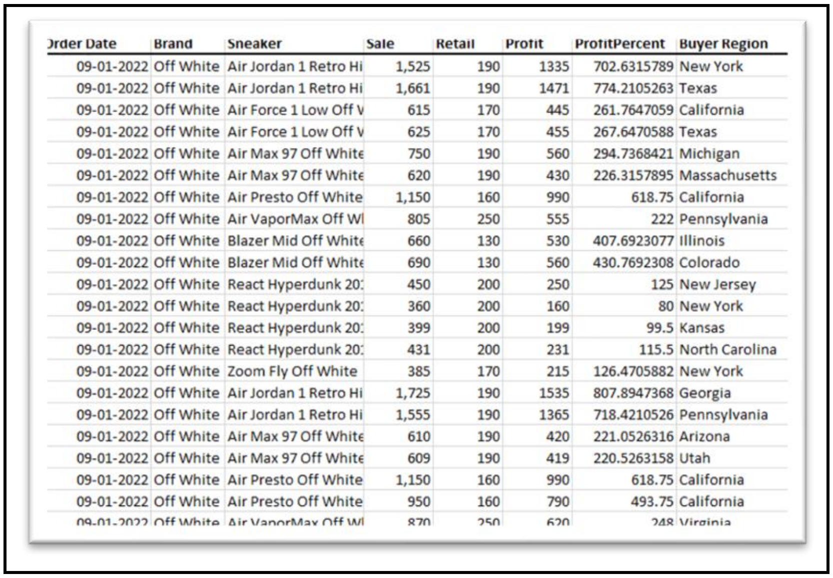
The "tFileList" component in Talend Open Studio is used to read a list of files from a directory and process them one by one. It is typically used in data integration jobs that need to process large amounts of files in a batch manner.

Here's a high-level overview of how the "tFileList" component works:

1. The "tFileList" component reads the file list from a directory and generates a row for each file.
2. The "tFileList" component passes the file name, path, and other information about the file to the next component in the data flow.
3. The next component in the data flow processes the file and produces the output.
4. The "tFileList" component then moves on to the next file in the list and repeats the process until all files have been processed.
5. The "tFileList" component provides various configuration options to control the processing of files, such as filtering files based on file name, size, date, or other criteria, and processing files in a specific order.

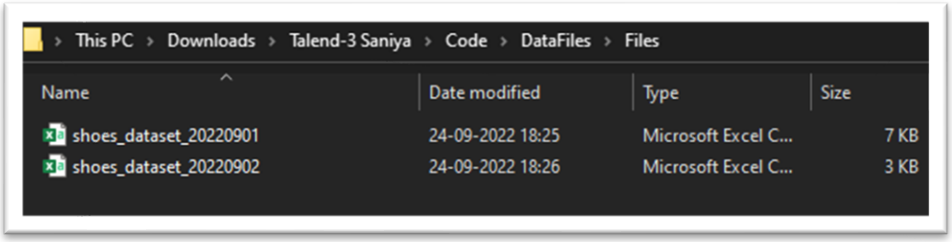
By using the "tFileList" component in combination with other components, you can easily automate complex file processing tasks and build efficient, scalable data integration jobs in Talend Open Studio.

One of the data files looks like this:



Ɔrder Date	Brand	Sneaker	Sale	Retail	Profit	ProfitPercent	Buyer Region
09-01-2022	Off White	Air Jordan 1 Retro Hi	1,525	190	1335	702.6315789	New York
09-01-2022	Off White	Air Jordan 1 Retro Hi	1,661	190	1471	774.2105263	Texas
09-01-2022	Off White	Air Force 1 Low Off V	615	170	445	261.7647059	California
09-01-2022	Off White	Air Force 1 Low Off V	625	170	455	267.6470588	Texas
09-01-2022	Off White	Air Max 97 Off White	750	190	560	294.7368421	Michigan
09-01-2022	Off White	Air Max 97 Off White	620	190	430	226.3157895	Massachusetts
09-01-2022	Off White	Air Presto Off White	1,150	160	990	618.75	California
09-01-2022	Off White	Air VaporMax Off Wl	805	250	555	222	Pennsylvania
09-01-2022	Off White	Blazer Mid Off White	660	130	530	407.6923077	Illinois
09-01-2022	Off White	Blazer Mid Off White	690	130	560	430.7692308	Colorado
09-01-2022	Off White	React Hyperdunk 20:	450	200	250	125	New Jersey
09-01-2022	Off White	React Hyperdunk 20:	360	200	160	80	New York
09-01-2022	Off White	React Hyperdunk 20:	399	200	199	99.5	Kansas
09-01-2022	Off White	React Hyperdunk 20:	431	200	231	115.5	North Carolina
09-01-2022	Off White	Zoom Fly Off White	385	170	215	126.4705882	New York
09-01-2022	Off White	Air Jordan 1 Retro Hi	1,725	190	1535	807.8947368	Georgia
09-01-2022	Off White	Air Jordan 1 Retro Hi	1,555	190	1365	718.4210526	Pennsylvania
09-01-2022	Off White	Air Max 97 Off White	610	190	420	221.0526316	Arizona
09-01-2022	Off White	Air Max 97 Off White	609	190	419	220.5263158	Utah
09-01-2022	Off White	Air Presto Off White	1,150	160	990	618.75	California
09-01-2022	Off White	Air Presto Off White	950	160	790	493.75	California
09-01-2022	Off White	Air VaporMax Off Wl	870	250	620	248	Virginia

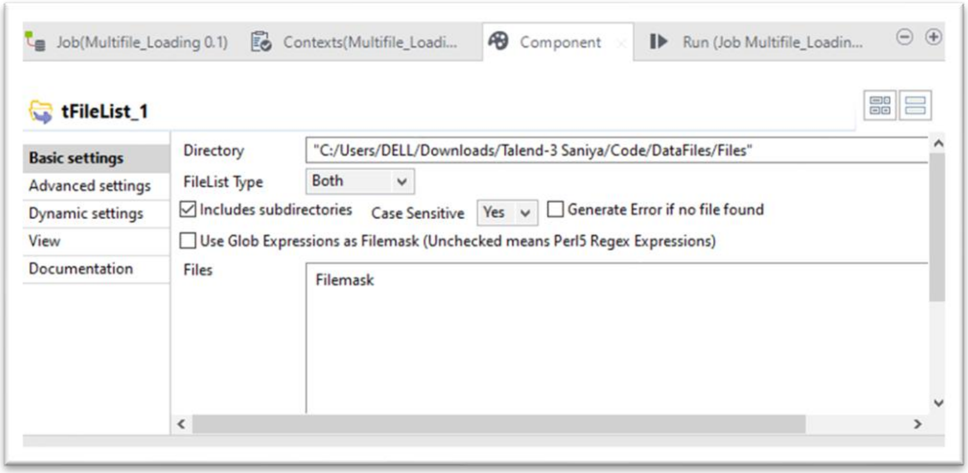
The files are stored in the 'Files' folder at this location:



This PC › Downloads › Talend-3 Saniya › Code › DataFiles › Files

Name	Date modified	Type	Size
shoes_dataset_20220901	24-09-2022 18:25	Microsoft Excel C...	7 KB
shoes_dataset_20220902	24-09-2022 18:26	Microsoft Excel C...	3 KB

The configuration of **tFileList_1** component is as follows:
Specify the path to the 'Files' folder in the Directory field.


Job(Multifile_Loading 0.1)
Contexts(Multifile_Loadi...
Component
Run (Job Multifile_Loadin...
Basic settings
Advanced settings
Dynamic settings
View
Documentation

tFileList_1

Directory "C:/Users/DELL/Downloads/Talend-3 Saniya/Code/DataFiles/Files"

FileList Type Both

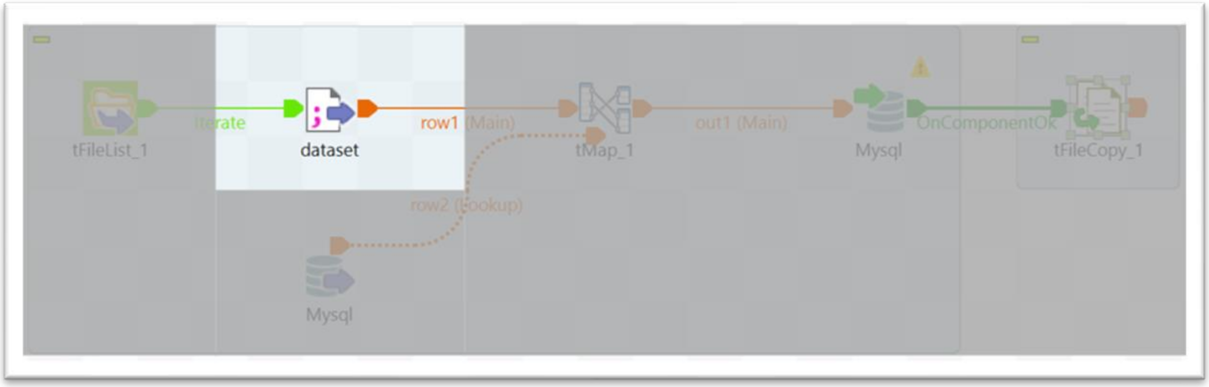
☑ Includes subdirectories Case Sensitive Yes □ Generate Error if no file found

□ Use Glob Expressions as Filemask (Unchecked means Perl5 Regex Expressions)

Files

Filemask

Dataset (tFileInputDelimited)



The tFileInputDelimited is a Talend component used to read data from delimited text files, such as CSV files. It is part of the Talend Open Studio for Data Integration software.

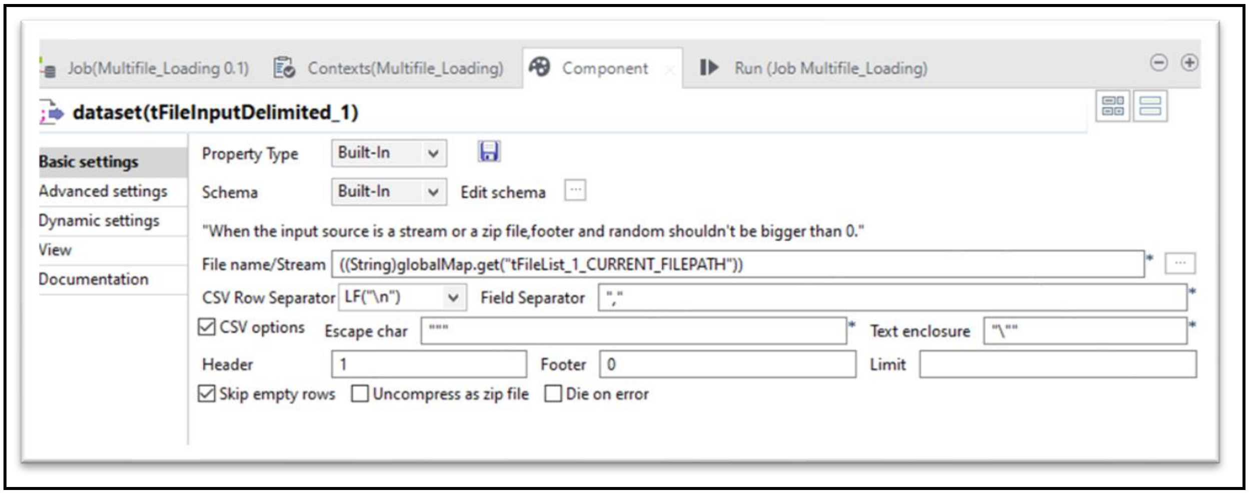
The tFileInputDelimited allows you to configure the delimiter used to separate values in your text file, as well as the text enclosing character used to encapsulate values that contain the delimiter. You can also specify the encoding used by your text file and the schema of your data, including the data types and names of each column.

Once configured, tFileInputDelimited can be used to read data from a delimited text file and extract the values into columns, which can then be transformed and loaded into a target database or data store.

In addition to reading data from delimited text files, tFileInputDelimited also provides a range of advanced features, such as processing large files in parallel, reading specific ranges of rows, and reading header rows.

In conclusion, tFileInputDelimited is a powerful and flexible Talend component that makes it easy to read data from delimited text files, such as CSV files. With its intuitive interface and advanced features, tFileInputDelimited is a valuable tool for data integration and ETL processes.

The configuration of **Dataset (tFileInputDelimited)** component is as follows:


Job(Multifile_Loading 0.1)
Contexts(Multifile_Loading)
Component
Run (Job Multifile_Loading)
Basic settings
Advanced settings
Dynamic settings
View
Documentation

dataset(tFileInputDelimited_1)

Property Type: Built-In

Schema: Built-In Edit schema

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

File name/Stream: ((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))

CSV Row Separator: LF("\n") Field Separator: ","

☑ CSV options Escape char: """" Text enclosure: "\""

Header: 1 Footer: 0 Limit:

☑ Skip empty rows □ Uncompress as zip file □ Die on error

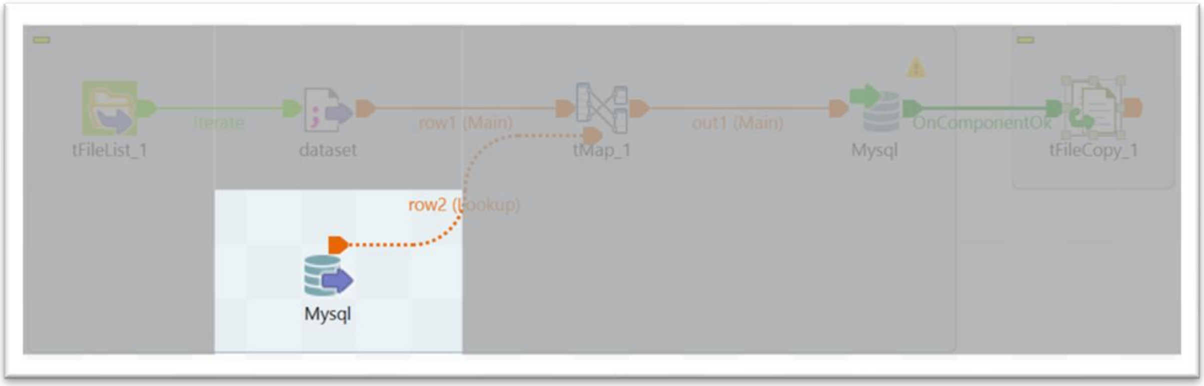
We pass the tFileList_1_CURRENT_FILEPATH to the File name field. This component reads all the csv file present in folder coming from previous component which is tFileList_1.

"tFileList_1_CURRENT_FILEPATH" is a dynamic column generated by the Talend component "tFileList".

"tFileList" is used to list the files in a directory, based on specified criteria, such as the file name, size, date, and type. The "tFileList_1_CURRENT_FILEPATH" column is a dynamic column that contains the file path of the current file being processed by the "tFileList" component.

"tFileList_1_CURRENT_FILEPATH" can be used as input for other components in the Talend job flow, such as "tFileInputDelimited" or "tFileInputJSON", to dynamically read the contents of the files in the directory.

tDBInput (MySQL)



tDBInput (MySQL) is a Talend component used to read data from a MySQL database. It is part of the Talend Open Studio for Data Integration software.

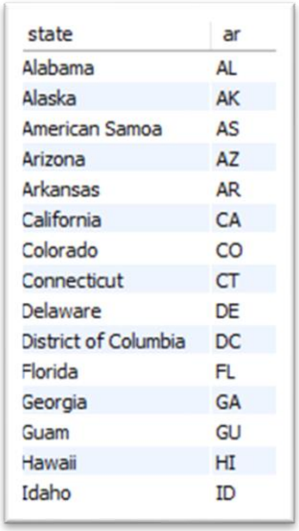
tDBInput (MySQL) allows you to connect to a MySQL database, specify a SQL query to retrieve data, and extract the data into columns, which can then be transformed and loaded into a target database or data store.

Before using tDBInput (MySQL), you need to configure the connection to your MySQL database, including the host name, port, database name, user name, and password. You can also specify advanced settings, such as the query timeout and fetch size.

Once connected, tDBInput (MySQL) can be used to retrieve data from the MySQL database, either in a batch mode or incrementally. In batch mode, tDBInput (MySQL) retrieves all the data in a single query, while in incremental mode, tDBInput (MySQL) retrieves data based on a specified key column, allowing you to process only the data that has changed since the last time the job was run.

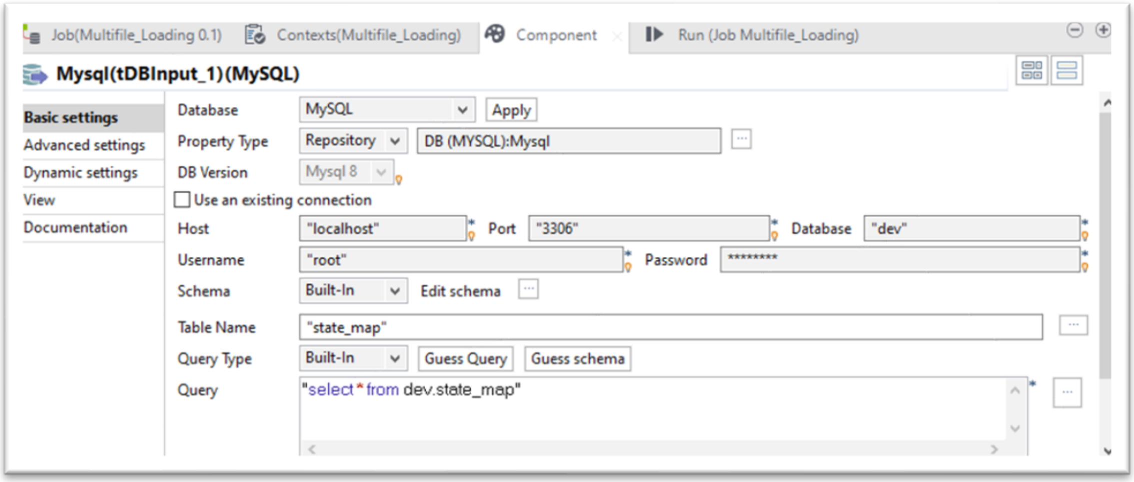
In conclusion, tDBInput (MySQL) is a powerful and flexible Talend component that makes it easy to read data from a MySQL database. With its intuitive interface and advanced features, tDBInput (MySQL) is a valuable tool for data integration and ETL processes that involve reading data from a MySQL database.

The ‘state_map’ table stored in MySQL looks like this:



state	ar
Alabama	AL
Alaska	AK
American Samoa	AS
Arizona	AZ
Arkansas	AR
California	CA
Colorado	CO
Connecticut	CT
Delaware	DE
District of Columbia	DC
Florida	FL
Georgia	GA
Guam	GU
Hawaii	HI
Idaho	ID

The configuration of **tDBInput (MySQL)** component is as follows:

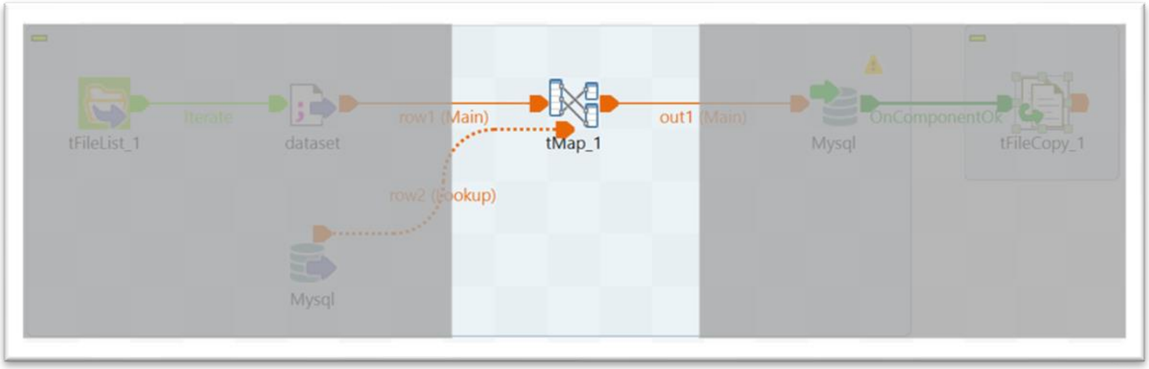


Mysql(tDBInput_1)(MySQL)

Database: MySQL
Property Type: Repository — DB (MYSQL):Mysql
DB Version: Mysql 8
□ Use an existing connection
Host: "localhost" Port: "3306" Database: "dev"
Username: "root" Password: ********
Schema: Built-In
Table Name: "state_map"
Query Type: Built-In
Query: "select * from dev.state_map"

We provide the Host, Port, Username, Password, Database, Table name, and Query.
This component will fetch all rows from ‘state_map’ table from MySQL Database and pass it to the next component, which is tMap_1.

tMap



The tMap component in Talend Open Studio is a powerful data transformation component used to manipulate and transform data records. The tMap component takes as input one or more flows of data records and generates an output flow of data records, based on the transformations and manipulations specified in the tMap configuration. The following are some of the common configurations in the tMap component:

1. **Input and Output:** In the tMap component, you can add multiple inputs and outputs to the mapping process. Inputs are the data sources that provide the data to be transformed and outputs are the targets that receive the transformed data.
2. **Data Mapping:** The tMap component provides a graphical interface for mapping the data from inputs to outputs. You can define the data mapping rules by selecting the columns from the inputs and linking them to the columns in the outputs. You can also apply transformations and calculations to the data as it is mapped.
3. **Joining Data:** The tMap component supports different types of joins, including inner join, left join, right join, and full outer join. You can join data from multiple inputs based on common columns to create a new output with combined data.
4. **Conditional Processing:** The tMap component allows you to apply conditions to the data mapping process. For example, you can filter out certain rows based on specific conditions, such as removing rows with null values, or splitting data into multiple outputs based on conditions.
5. **Error Handling:** The tMap component provides error handling options to handle exceptions that may occur during the data mapping process. You can define error handling rules to catch errors, such as logging the errors to a file or routing the erroneous rows to a separate output.
6. **Reusable Transformations:** The tMap component provides the ability to reuse transformations, allowing you to define complex data transformations once and reuse them in multiple jobs.

In conclusion, the tMap component in Talend is a highly configurable and flexible tool for transforming and manipulating data. By leveraging its various configurations, you can efficiently map data from multiple sources to desired outputs, perform complex data transformations, and handle errors and exceptions.

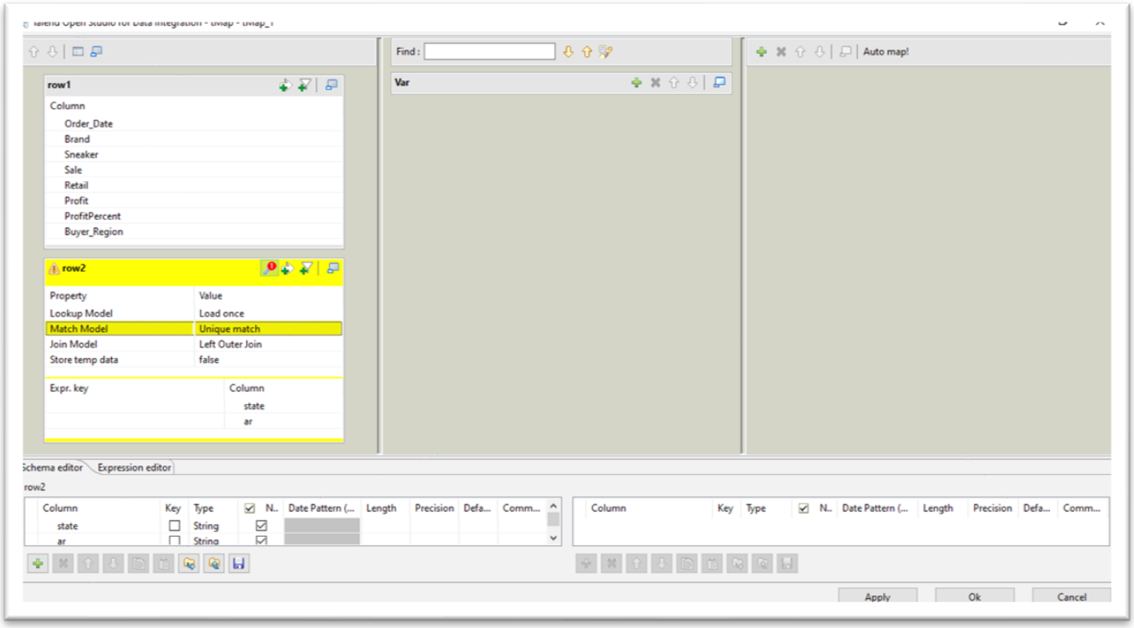
The "tMap" component in Talend supports two types of inputs: Main Input and Lookup Input.

1. **Main Input:** The Main Input is the primary input that provides the data to be transformed. The data from the Main Input is processed by the tMap component and can be mapped to multiple outputs.
2. **Lookup Input:** The Lookup Input is a secondary input that provides supplementary data to be used in conjunction with the Main Input data. The tMap component can use the data from the Lookup Input to enrich the data from the Main Input. For example, you can use the Lookup Input to lookup reference data, such as a lookup table, and join it with the Main Input data based on a common key.

The tMap component supports several types of Lookup Inputs, including inner join, left join, right join, and full outer join. The Lookup Inputs can be used to filter, aggregate, and manipulate the Main Input data.

We need to pass the output of previous components to tMap component. In order to pass input of the **Dataset (tFileInputDelimited)** component to tMap, right-click on the dataset, select Row, select Main and connect it to tMap. Follow the same steps for **tDBInput (MySQL)** component. We can see that dataset is our Main Input, and tDBInput is our Lookup Input.

When we double click on tMap component, you will see the following window:



Talend Open Studio for Data Integration - tMap - tMap_1

row1
Column
Order_Date
Brand
Sneaker
Sale
Retail
Profit
ProfitPercent
Buyer_Region

row2

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

Expr. key	Column
	state
	ar

row2

Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...
state	☐	String	☑					
ar	☐	String	☑					

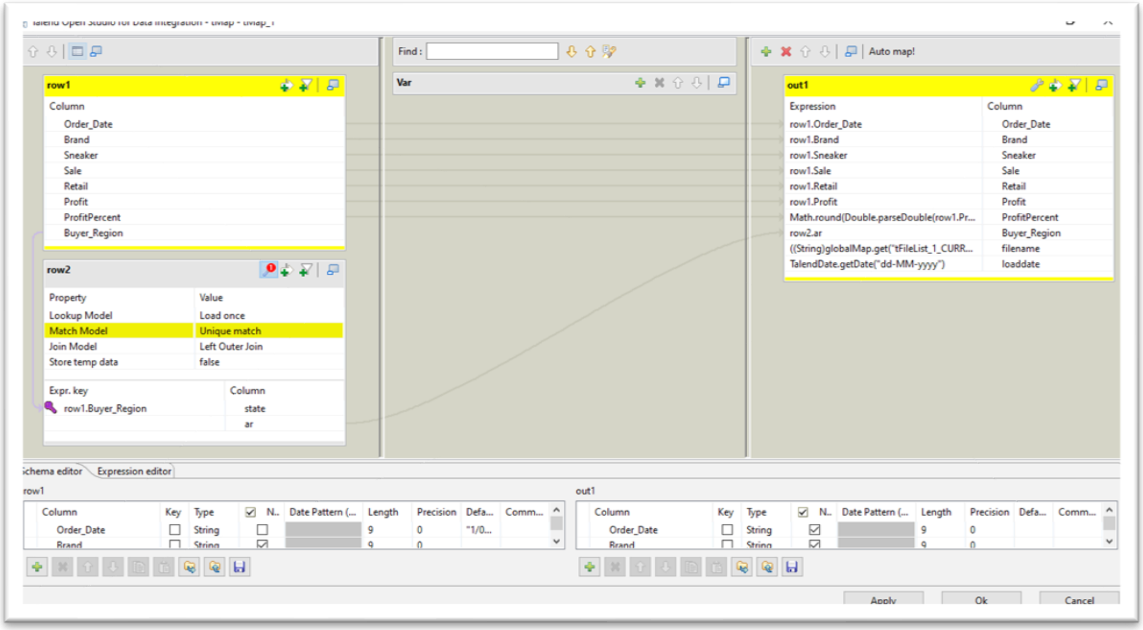
Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...

We can see there are 2 inputs. The main input is data from the delimited file, and the lookup input is the data coming from the MySQL table.

Buyer_Region is the foreign key. We will use the Buyer_Region column to join two tables. To join two tables, drag Buyer_Region from row1 and put it to Expr. key of row2.

We need to create a final table to store the transformed data. For that in the right most rectangle, click on plus icon which says Add an Output table, give it a name and click ok.

It will create an output table. Now drag and drop the columns from row1 and row2 tables to the final output table.



row1
Column
Order_Date
Brand
Sneaker
Sale
Retail
Profit
ProfitPercent
Buyer_Region

row2

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

Expr. key	Column
row1.Buyer_Region	state
	ar

out1

Expression	Column
row1.Order_Date	Order_Date
row1.Brand	Brand
row1.Sneaker	Sneaker
row1.Sale	Sale
row1.Retail	Retail
row1.Profit	Profit
Math.round(Double.parseDouble(row1.Pr...	ProfitPercent
row2.ar	Buyer_Region
((String)globalMap.get("tFileList_1_CURR...	filename
TalendDate.getDate("dd-MM-yyyy")	loaddate

row1

Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...
Order_Date	□	String	□		9	0	"1/0...	
Brand	□	String	☑		9	0		

out1

Column	Key	Type	☑ N..	Date Pattern (...	Length	Precision	Defa...	Comm...
Order_Date	□	String	☑		9	0		
Brand	□	String	☑		9	0		

We have performed a transformation on the 'ProfitPercent' column. We have rounded up the value to 2 decimals. Use the following formula:

```
Math.round(Double.parseDouble(row1.ProfitPercent) * 100.0)/100.0
```

We have added two more columns.
The filename contains the name of the processed file. Use the following formula to get the file name:

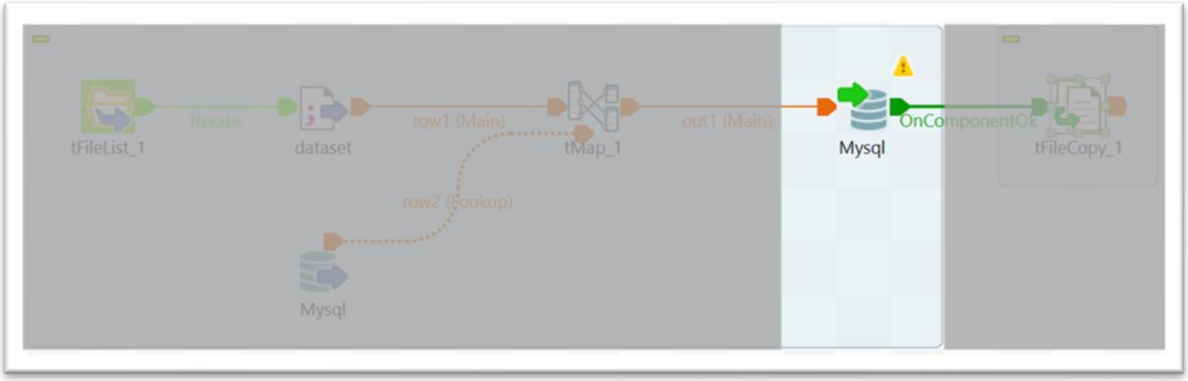
```
((String)globalMap.get("tFileList_1_CURRENT_FILE"))
```

Load date column contains the date at which the file was processed. Use the following formula to get the current date:

```
TalendDate.getDate("dd-MM-yyyy")
```

At the end, we will have a final output table ready with the transformed columns. We will send this data to the next component which is **tDBOutput (MySQL)** to write the data in MySQL Database.

tDBOutput (MySQL)



The tDBOutput (MySQL) is a Talend component used to write data to a MySQL database. It is part of the Talend Open Studio for Data Integration software.

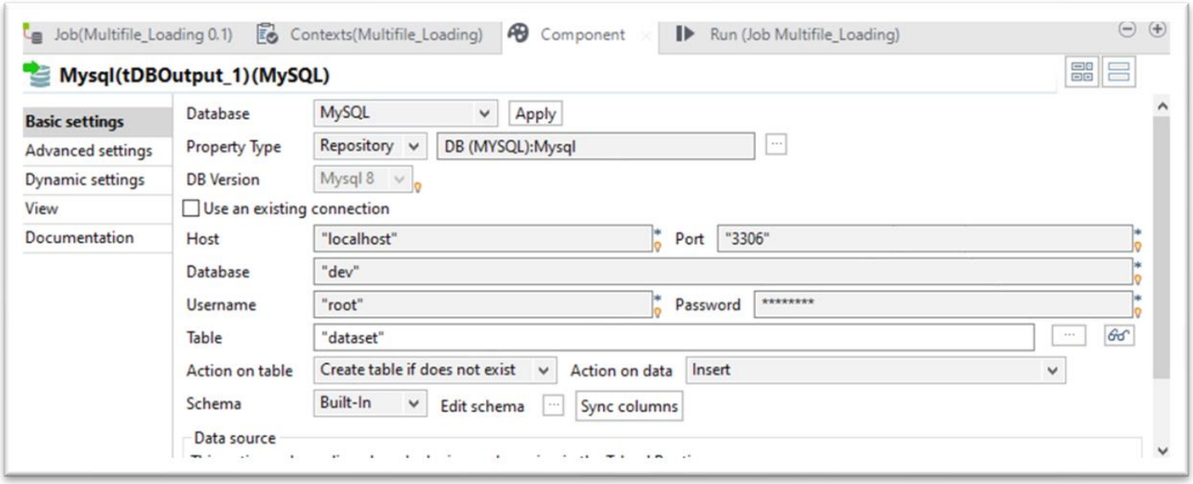
The tDBOutput (MySQL) allows you to connect to a MySQL database, specify the target table, and map the columns from your data to the columns in the target table. You can also configure the insert method, including insert, update, and upsert, which determines how data is written to the target table.

Before using tDBOutput (MySQL), you need to configure the connection to your MySQL database, including the host name, port, database name, user name, and password. You can also specify advanced settings, such as the batch size, commit size, and query timeout.

Once configured, tDBOutput (MySQL) can be used to write data to the target table in the MySQL database. You can use tDBOutput (MySQL) in batch mode to write all the data in a single transaction or in stream mode to write the data incrementally.

In conclusion, tDBOutput (MySQL) is a powerful and flexible Talend component that makes it easy to write data to a MySQL database. With its intuitive interface and advanced features, tDBOutput (MySQL) is a valuable tool for data integration and ETL processes that involve writing data to a MySQL database.

The configuration of the **tDBOutput(MySQL)** component looks like this:

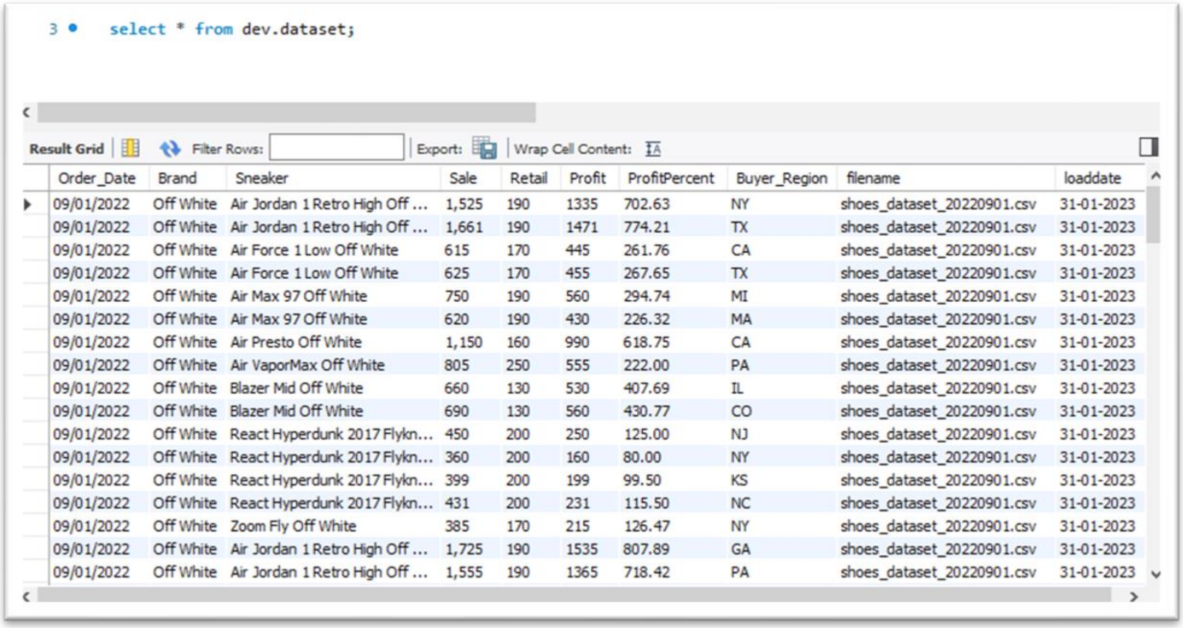


Mysql(tDBOutput_1)(MySQL)

Database: MySQL
Property Type: Repository | DB (MYSQL):Mysql
DB Version: Mysql 8
☐ Use an existing connection
Host: "localhost" Port: "3306"
Database: "dev"
Username: "root" Password: ********
Table: "dataset"
Action on table: Create table if does not exist Action on data: Insert
Schema: Built-In

We provide the Host, Port, Username, Password, Database, Table name, and Query.
This component will write data from the previous component tMap_1 to the 'dataset' table in MySQL Database.

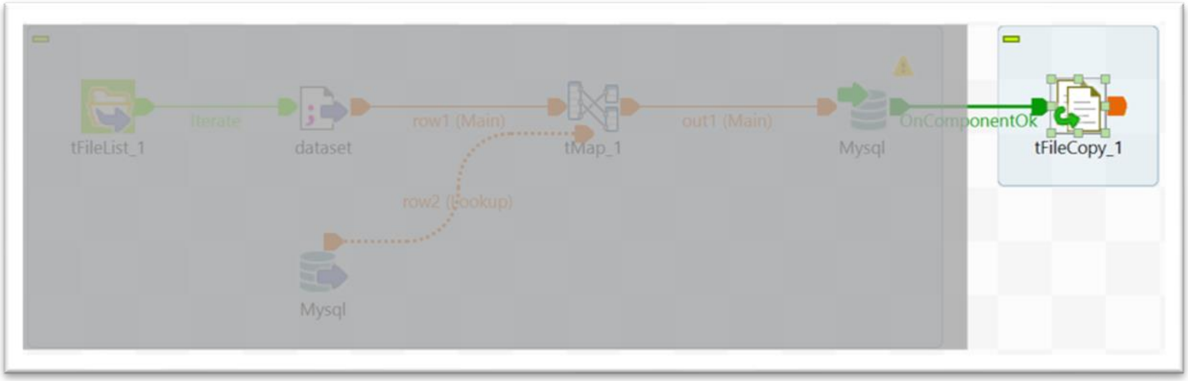
The 'dataset' table in the MySQL Database looks like this:



```
3 •   select * from dev.dataset;
```

Order_Date	Brand	Sneaker	Sale	Retail	Profit	ProfitPercent	Buyer_Region	filename	loaddate
09/01/2022	Off White	Air Jordan 1 Retro High Off ...	1,525	190	1335	702.63	NY	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Jordan 1 Retro High Off ...	1,661	190	1471	774.21	TX	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Force 1 Low Off White	615	170	445	261.76	CA	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Force 1 Low Off White	625	170	455	267.65	TX	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Max 97 Off White	750	190	560	294.74	MI	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Max 97 Off White	620	190	430	226.32	MA	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Presto Off White	1,150	160	990	618.75	CA	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air VaporMax Off White	805	250	555	222.00	PA	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Blazer Mid Off White	660	130	530	407.69	IL	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Blazer Mid Off White	690	130	560	430.77	CO	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	React Hyperdunk 2017 Flykn...	450	200	250	125.00	NJ	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	React Hyperdunk 2017 Flykn...	360	200	160	80.00	NY	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	React Hyperdunk 2017 Flykn...	399	200	199	99.50	KS	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	React Hyperdunk 2017 Flykn...	431	200	231	115.50	NC	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Zoom Fly Off White	385	170	215	126.47	NY	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Jordan 1 Retro High Off ...	1,725	190	1535	807.89	GA	shoes_dataset_20220901.csv	31-01-2023
09/01/2022	Off White	Air Jordan 1 Retro High Off ...	1,555	190	1365	718.42	PA	shoes_dataset_20220901.csv	31-01-2023

tFileCopy



The tFileCopy component in Talend Open Studio is used to copy files from one location to another. The tFileCopy component takes as input the source file path and the target file path, and performs a file copy operation from the source to the target.

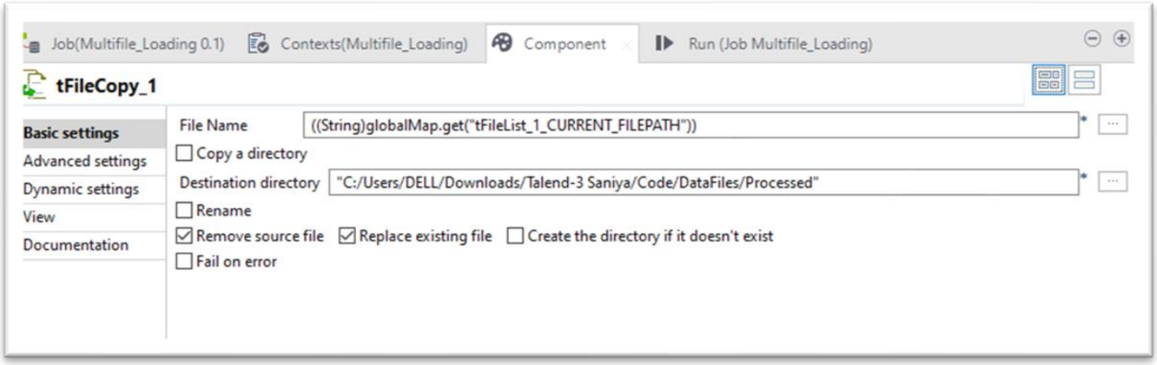
The tFileCopy component supports various file copy modes, such as file overwrite, file append, and file merge. You can specify the file copy mode using the component's configuration options.

In addition to basic file copy, the tFileCopy component provides several advanced options, such as the ability to filter files based on file name patterns, to preserve file attributes, and to perform a file copy operation only if the target file does not exist or is older than the source file.

The tFileCopy component is a powerful tool for file management, and can be used in various file-related scenarios, such as file backup, file synchronization, file archiving, and more. By using the tFileCopy component in combination with other components in Talend Open Studio, you can build efficient and scalable file management solutions that meet your specific requirements.

In conclusion, the tFileCopy component in Talend Open Studio is a flexible and versatile tool for file copy and management. With its advanced options and powerful features, the tFileCopy component is ideal for a wide range of file management scenarios.

The configuration of the **tFileCopy_1** component looks like this:


Job(Multifile_Loading 0.1)
Contexts(Multifile_Loading)
Component
Run (Job Multifile_Loading)
Basic settings
Advanced settings
Dynamic settings
View
Documentation

tFileCopy_1

File Name ((String)globalMap.get("tFileList_1_CURRENT_FILEPATH"))

□ Copy a directory

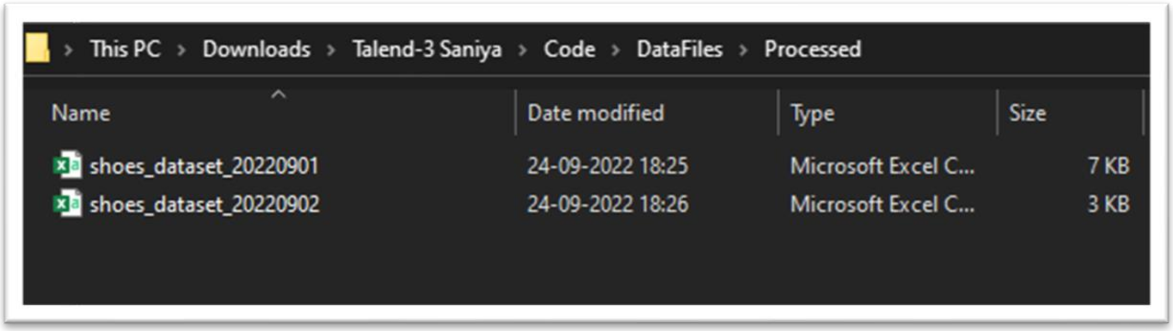
Destination directory "C:/Users/DELL/Downloads/Talend-3 Saniya/Code/DataFiles/Processed"

□ Rename

☑ Remove source file ☑ Replace existing file □ Create the directory if it doesn't exist

□ Fail on error

This component will copy the files from the 'Files' folder to the 'Processed' folder.
The 'Processed' folder contains the following files after the successful execution of the pipeline:



› This PC › Downloads › Talend-3 Saniya › Code › DataFiles › Processed

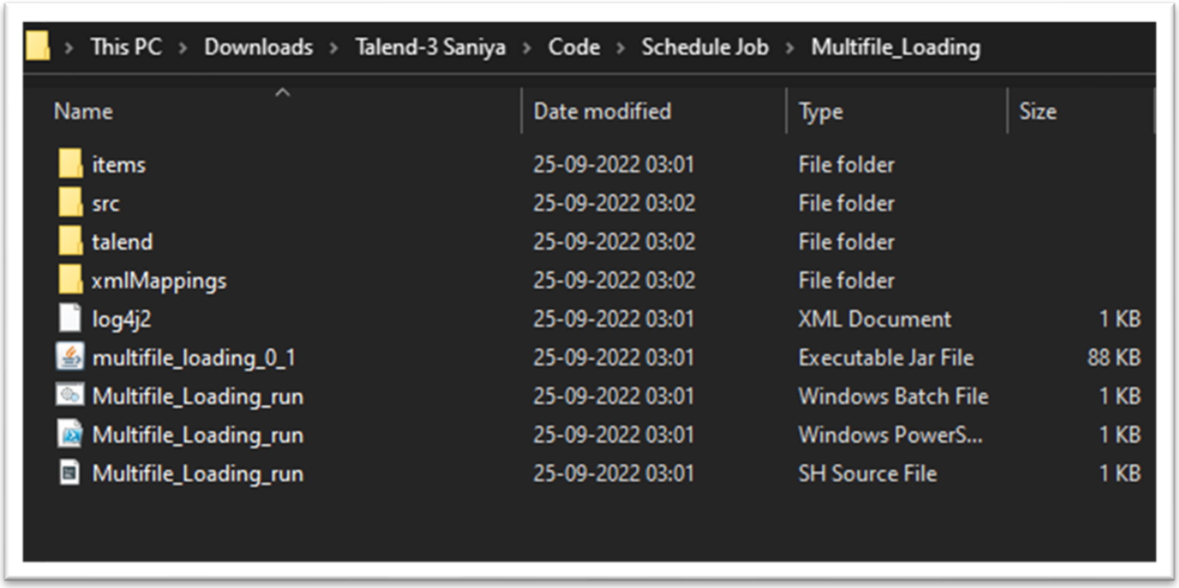
Name	Date modified	Type	Size
shoes_dataset_20220901	24-09-2022 18:25	Microsoft Excel C...	7 KB
shoes_dataset_20220902	24-09-2022 18:26	Microsoft Excel C...	3 KB

Build a Talend Job

To build a job in Talend Open Studio, follow these steps:

1. Open your Talend Open Studio project and select the job you want to build.
2. Right-click on the job and select "Build Job" from the context menu.
3. In the Build Job wizard, select the destination folder where you want to save the job.
4. Click "Finish" to build and export the job.

The exported job will be saved in the specified destination folder in your selected format. It will create a zip folder. Once you unzip, you will see the following files:



This PC › Downloads › Talend-3 Saniya › Code › Schedule Job › Multifile_Loading

Name	Date modified	Type	Size
items	25-09-2022 03:01	File folder	
src	25-09-2022 03:02	File folder	
talend	25-09-2022 03:02	File folder	
xmlMappings	25-09-2022 03:02	File folder	
log4j2	25-09-2022 03:01	XML Document	1 KB
multifile_loading_0_1	25-09-2022 03:01	Executable Jar File	88 KB
Multifile_Loading_run	25-09-2022 03:01	Windows Batch File	1 KB
Multifile_Loading_run	25-09-2022 03:01	Windows PowerS...	1 KB
Multifile_Loading_run	25-09-2022 03:01	SH Source File	1 KB

We will use the Windows Batch File to schedule this job on the Task scheduler.

Schedule the job using Windows Task Scheduler

To schedule a Windows batch file using Windows Task Scheduler, follow these steps:

1. Open the Windows Task Scheduler by pressing the Windows key + R, typing "taskschd.msc" and hit Enter.
2. Click "Create Basic Task" from the Actions panel on the right side.
3. In the Create Basic Task wizard, provide a name and description for the task.
4. Select the trigger for the task. You can run the task daily, weekly, monthly, or when the computer starts.
5. Select the start time for the task and any other relevant options, such as the end time or the interval between repeats.
6. In the Action section, select "Start a program" and click "Next".
7. In the Program/script field, enter the path to the batch file, e.g., 'C:\Project\Multifile_Loading_run.bat'.
8. Click "Next" and "Finish" to complete the task setup.
9. The task will run at the specified schedule and execute the batch file.

Summary

- A Talend Real-Time Project for ETL (Extract, Transform, Load) Process Automation is a project designed to automate the process of extracting data from various sources, transforming it into a desired format, and loading it into a target system, such as a data warehouse. The Talend Open Studio is used to develop and execute the ETL job, which can handle large volumes of data and process it in real-time.
- The prerequisites for this project include a basic understanding of data warehousing concepts, experience with SQL, and knowledge of Talend Open Studio components, such as tMap, tMySQLInput, and tMySQLOutput.
- The Talend job can be built using various components, such as tFileInputDelimited for reading data from a file, tDBInput for reading data from a database, and tMap for transforming data from one format to another. The transformed data is then loaded into the target system using tDBOutput or tMySQLOutput.
- The job can be exported as a standalone Java application or a command-line executable, and scheduled to run automatically using Windows Scheduler.
- The end result of this project is a streamlined ETL process that saves time and reduces the risk of errors, leading to increased productivity and improved data quality.