tPostgresqlTableTransfer

Purpose

This component inserts records from one table into another table. It generates an internal field mapping, by matching equal column names in both tables.
If there are columns which should not be transferred (e.g. because they are auto increment columns), you can exclude them from the transfer with the Exclude fields section.
If the target table expects more columns filled with fixed values, you can add them in the Fixed Column Value section.
By using two asynchronous threads, the component gains more performance because the component can read and write at the same time.
It's main working scenario is copying data from one database to another or create backup files.

Talend-Integration

This component can be found in the palette under Database -> PostgreSQL

Basic Settings

Property	Content
Use data source	Check this if the component should use a connection from a database connection pool. Otherwise the component expects external connection components.
Source data source alias	Set here the name of the database pool for the source connection.
Target data source alias	Set here the name of the database pool for the target connection.
Source Connection	The connection used to read the data. This is not necessarily a MySQL connection. The component can also use connections from other databases. If there are some mismatches between the data types, you can adjust the type mapping in the advanced settings in the Custom Type Mapping.
Target Connection	This must be a tPostgresqlConnection. This connection is only used if the option " Only backup in file, no inserts into table" is not set.
Self-defined query	Check this if the source data should be read from a query. Otherwise the component takes the data from the given source table
Source Table	Source table, if not a self-defined query is used.
Source table where clause	The where condition. A missing where keyword will be added automatically.
Truncate Table	Check this if the component have to truncate the table before import new data.
Enable exclude source columns	This enables the table of the columns to exclude
Source columns to exclude	Add here the names of the columns which have to be excluded. The generated query will not contain these columns. This works only if the query is generated from the component itself, if you provide your own query, this list will not be applied.
Enable columns with fixed values	This enables the table of fixed column values
Columns with fixed values	If the target table expect some columns with fixed valued, setup them here. The value expression is a java expression like context variables or literals.
Log Interval	Because of there is no flow visible in the job, the component sends a log message about the progress. Setup here how often this happened. This interval is also the interval in which the component detects the end of the processing or errors.
Die On Error	If there is something wrong while reading or writing data, the component will throw an error to the job.
Backup Data in File	If true, the component writes the source data also in a CSV-file. See the format information below.

Only backup in file, no inserts into table	If true, the component only writes the backup data and performes no inserts (and also no truncates etc.)
Backup file or directory	If it is a directory, the component use the target table name as file name +".csv" and if the entry points to a file, it takes exactly this to write into.
Export boolean as number	If true boolean values will exported in the file as 0 or 1 instead of false or true.

Format information about the backup file:

- The charset is UTF-8
- The line delimiter is in UNIX format
- Fields are separated by semicolon
- All values are quoted with double quote
- Line breaks in a field content will survive
- Already existing double quotes will be escaped with a backslash
- Already existing backslashes will also be escaped
- Null values are written as \N
- Boolean values are written per default als 0 or 1 (can be changed)
- Date, datetime or timestamp values are written in the format yyyy-MM-dd HH:mm:ss

This is a very common file format to transfer data. These kind of files can be directly used with the PostgreSQL bulk import.

Advanced Settings

Property	Content
Source select fetch size	The number records the component let the driver read at once. This can increase the performance.
Insert batch size	The component uses the batch mode and this is the number of records kept in memory until send to the database as batch request.
Logout query	If set, the component prints the select query to standard out.
Logout insert statement	If set, the component prints the insert statement to standard out. It is a prepared statement printed once.
Enable Log4J	Enables the internally used Log4J framework
Debug	In case of problems the component prints a lot of useful debugging information
Custom Type Mapping	Here you can setup in which way specific database types are mapped to the internal java types. All types not mentioned here will be read as Object and transferred to the target database as Object.
Reuse data model information for further runs	This option keeps the already collected information about schemas, tables and columns within the memory for further usage. This can shorten the necessary initialisation phase.
Alternative key to keep the data model in memory	The data model information will be hold by a key in the memory. By using the same key for different components you can share the same data model for different components. Normally it is supposed to keep this attribute empty. In this case the data model will be reused only for the same component.

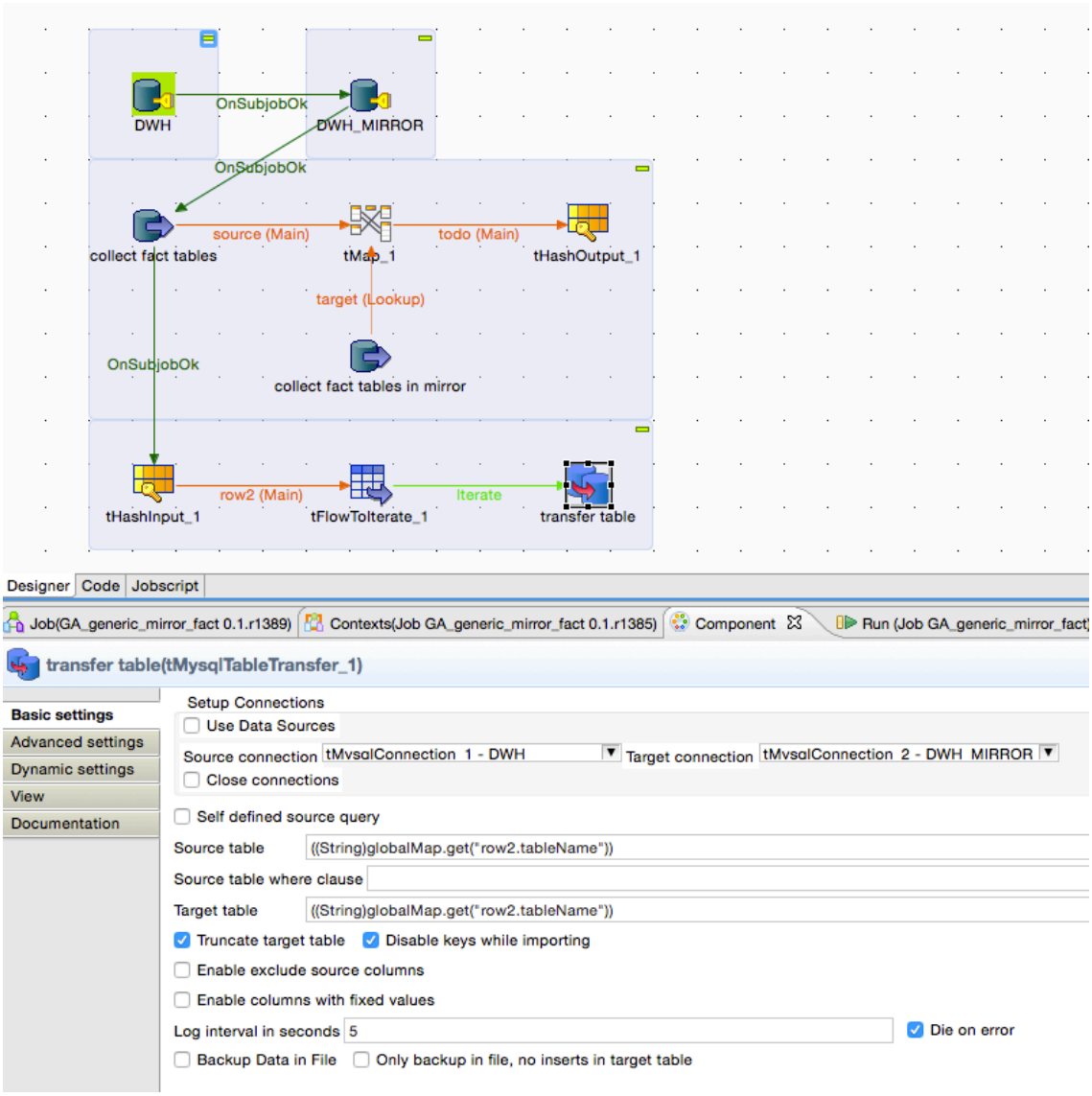
Return values

Return value	Content
ERROR_MESSAGE	Last error message
NB_LINE	Number rows read
NB_INSERTS	Number rows inserted into the target table
SOURCE_QUERY	Query used to select the source records

SOURCE_TABLE	Source table (not in case of using self-defined query)
TARGET_TABLE	Target table (not in case of using only backup mode)
BACKUP_FILE	The current written backup file with the full path.

Scenario 1: Copy tables from the actual database into a mirror database

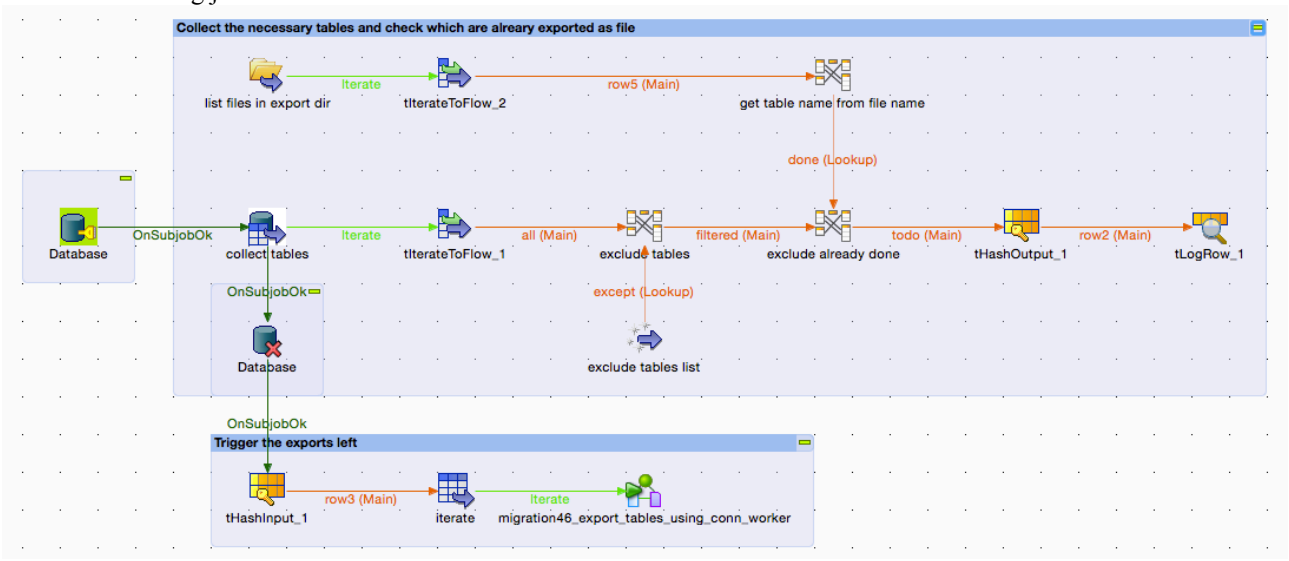
This job reads all table names from the DWH database and compares it with the table names from the mirror database and starts the transfer only for the tables existing in both databases. The example is actually from the component tMysqlTableTransfer but it works exactly in the same way.



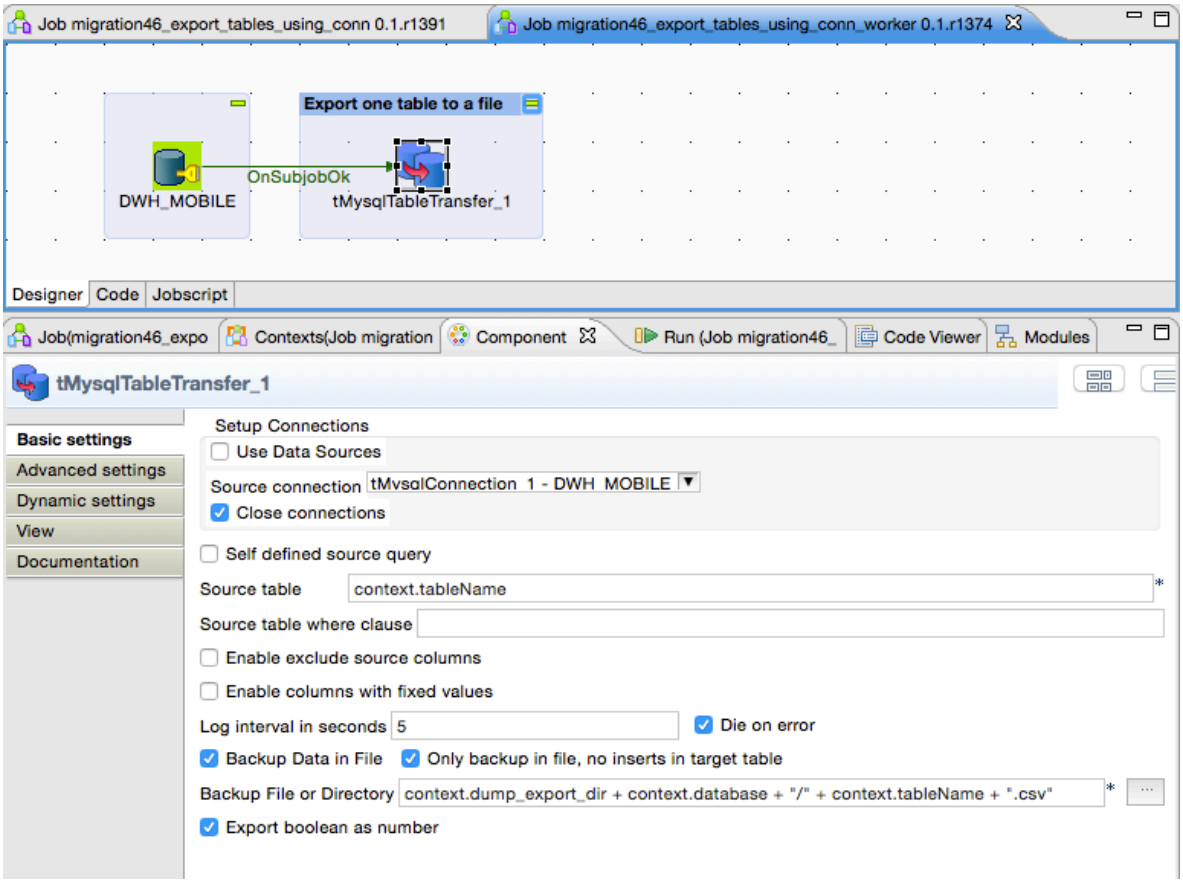
The table name is the same for source and target. The component takes both connections build at the start of the job and should therefore not close them because the component runs within an iteration.

Scenario 2: Backup all tables

In this scenario all tables will be simply dumped as CSV files but the job has to avoid dumping a table twice.
This is the steering job:



... and this is the simple worker job triggered here (the example shows the component tMysqlTableTransfer but it works in exactly the same way):



Because of the option "Only backup in file, no inserts in target table" we only need one connection and all settings for the target table are hidden.