Pentaho Data Integration

Spoon 3.0 User Guide

Last Modified on November 11th, 2007

1. Contents

2. About This Document

2.1. What it is

This document is a technical description of Spoon, the graphical transformation and job designer of the Pentaho Data Integration suite also known as the Kettle project.

2.2. What it is not

This document does not attempt to describe in great detail how to create jobs and transformations for all possible situations. Recognizing that different developers have different approaches to designing their data integration solutions, Spoon empowers users with the freedom and flexibility to design solutions in the manner they feel most appropriate to the problem at hand – and that is the way it should be!
Other documentation
Here are links to other documents that you might be interesting to go through when you are building transformations:

Flash demos, screen shots, and an introduction to building a simple transformation:
http://kettle.pentaho.org/screenshots/

Pentaho Data Integration community website – news, case studies, weekly tips and more:
http://kettle.pentaho.org

Pentaho Data Integration Forum – discussions on design, features, bugs and enhancements:
http://forums.pentaho.org/forumdisplay.php?f=69

Running transformations in batch using Pan: Pan-3.0.pdf

Running jobs in batch using Kitchen: Kitchen-3.0.pdf

An introduction to Pentaho Data Integration in Roland Bouman's blog:
http://rpbouman.blogspot.com/2006/06/pentaho-data-integration-kettle-turns.html

Nicholas Goodman is also blogging on Kettle and BI: http://www.nicholasgoodman.com

Here is a link to the centralized Pentaho issue tacking system: http://jira.pentaho.org

3. Introduction to Spoon

3.1. What is Spoon?

Kettle is an acronym for "Kettle E.T.T.L. Environment". This means it has been designed to help you with your ETTL needs: the Extraction, Transformation, Transportation and Loading of data.

Spoon is a graphical user interface that allows you to design transformations and jobs that can be run with the Kettle tools Pan and Kitchen. Pan is a data transformation engine that is capable of performing a multitude of functions such as reading, manipulating and writing data to and from various data sources. Kitchen is a program that can execute jobs designed by Spoon in XML or in a database repository. Usually jobs are scheduled in batch mode to be run automatically at regular intervals.

NOTE: *For a complete description of Pan or Kitchen, please refer to the Pan and Kitchen user guides.*

Transformations and Jobs can describe themselves using an XML file or can be put in a Kettle database repository. This information can then be read by Pan or Kitchen to execute the described steps in the transformation or run the job.

In short, Pentaho Data Integration makes data warehouses easier to build, update and maintain!

3.2. Installation

The first step is the installation of Sun Microsystems Java Runtime Environment version 1.4 or higher. You can download a JRE for free at http://www.javasoft.com/.

After this, you can simply unzip the zip-file: Kettle-3.0.0.zip in a directory of your choice. In the Kettle directory where you unzipped the file, you will find a number of files. Under Unix-like environments (Solaris, Linux, MacOS, …) you will need to make the shell scripts executable. Execute these commands to make all shell scripts in the Kettle directory executable:

```
cd Kettle
chmod +x *.sh
```

3.3. Launching Spoon

To launch Spoon on the different platforms these are the scripts that are provided:

Spoon.bat: launch Spoon on the Windows platform.
sh spoon.sh: launch Spoon on a Unix-like platform: Linux, Apple OSX, Solaris, ...

If you want to make a shortcut under the Windows platform an icon is provided: "spoon.ico" to set the correct icon. Simply point the shortcut to the Spoon.bat file.

3.4. Supported platforms

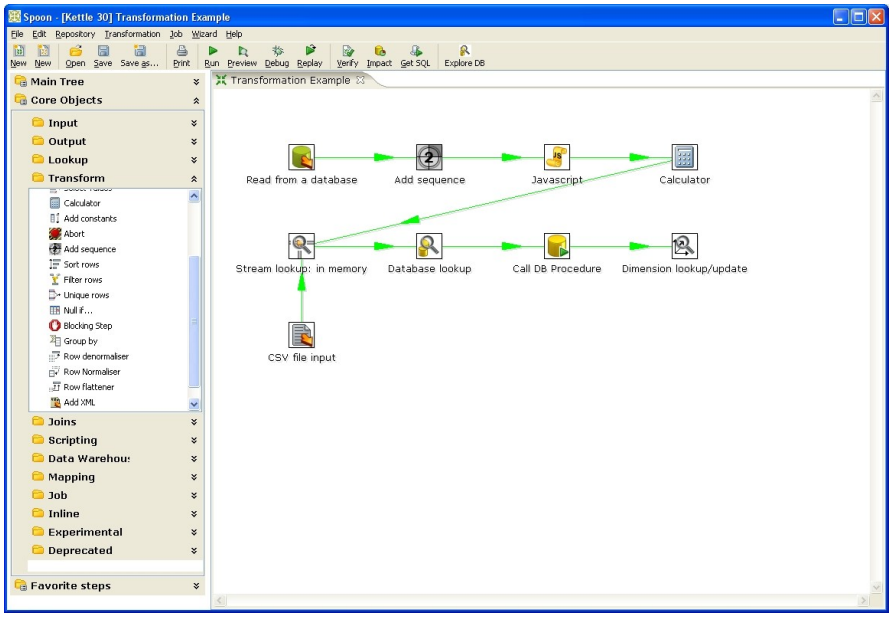
The Spoon GUI is supported on the following platforms:

- Microsoft Windows: all platforms since Windows 95, including Vista
- Linux GTK: on i386 and x86_64 processors
- Apple's OSX: works both on PowerPC and Intel machines
- Solaris: using a Motif interface (GTK optional)
- AIX: using a Motif interface
- HP-UX: using a Motif interface (GTK optional)

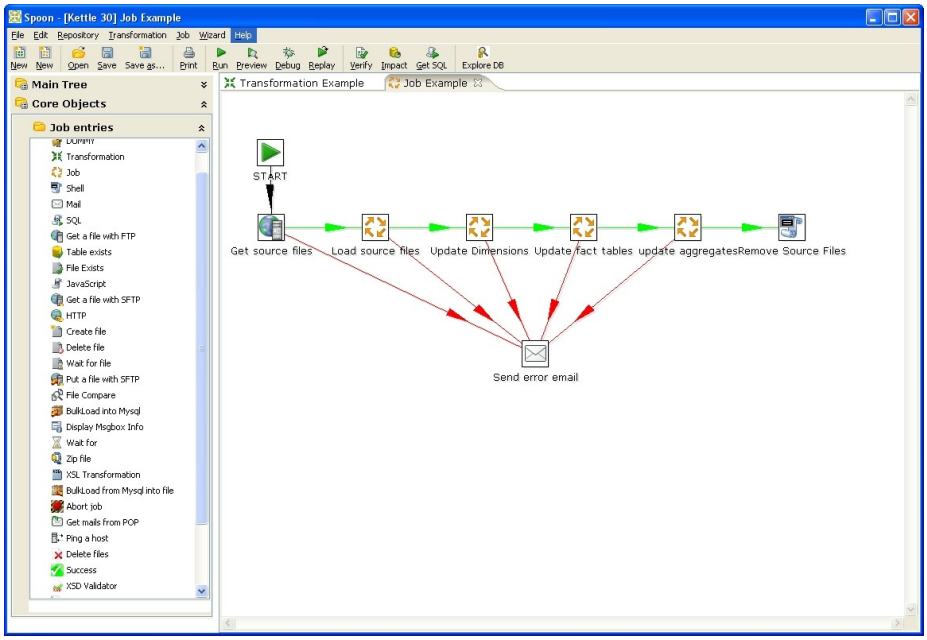
3.5. Known Issues

Please check the Tracker lists at http://jira.pentaho.org/browse/PDI for up-to-date information on discovered issues.

3.6. Screen shots



Designing a Transformation



Designing a job

The Main tree in the upper-left panel of Spoon allows you to browse connections along with the jobs and transformations you currently have open. When designing a transformation, the Core Objects palate in the lower left-panel contains the available steps used to build your transformation including input, output, lookup, transform, joins, scripting steps and more. When designing a job, the Core objects palate contains the available job entries. When designing a job, the Core Objects bar contains a variety of job entry types.

These items are described in detail in the chapters below: 4. Database Connections, 7. Hops, 10. Transformation Steps, 12. Job Entries, 13. Graphical View.

3.7. Command line options

These are the command line options that you can use when starting the Spoon application:

```
-file=filename
```

This option runs the specified transformation (.ktr : Kettle Transformation).

```
-log=Logging Filename
```

This option allows you to specify the location of the log file. The default is the standard output.

```
-level=Logging Level
```

The level option sets the log level for the transformation being run.
These are the possible values:

Nothing:	Do not show any output
Error:	Only show errors
Minimal:	Use minimal logging
Basic:	This is the default basic logging level
Detailed:	Give detailed logging output
Debug:	Show very detailed output for debugging purposes.
Rowlevel:	Detailed logging at a row level. *Warning - this will generate a lot of data.*

```
-rep=Repository name
```

Connect to the repository with name *"Repository name".*

Note: You also need to specify the options –user, –pass and –trans described below. The repository details are loaded from the file repositories.xml in the local directory or in the Kettle directory:

$HOME/.kettle/ or C:\Documents and Settings\<username>\.kettle on Windows.

```
-user=Username
```

This is the username with which you want to connect to the repository.

```
-pass=Password
```

The password to use to connect to the repository.-trans=*Transformation Name*

Use this option to select the transformation to run from the repository.

```
-job=Job Name
```

Use this option to select the job to run from the repository.

Important Notes:

- *On Windows, we advise you to use the `/option:value` format to avoid command line parsing problems by the MS-DOS shell.*

Fields in italic represent the values that the options use.

- *It's important that if spaces are present in the option values, you use quotes or double quotes to keep them together. Take a look at the examples below for more info.*

3.8. Repository

Spoon provides you with the ability to store transformation and job files to the local file system or in the Kettle repository. The Kettle repository can be housed in any common relational database. This means that in order to load a transformation from a database repository, you need to connect to this repository. To do this, you need to define a database connection to this repository. You can do this using the repositories dialog you are presented with when you start up Spoon:

The Repository login screen

The information concerning repositories is stored in a file called "repositories.xml". This file resides in the hidden directory ".kettle" in your default home directory. On windows this is C:\Documents and Settings\<username>\.kettle

Note: *The complete path and filename of this file is displayed on the Spoon console.*

If you don't want this dialog to be shown each time Spoon starts up, you can disable it by unchecking the 'Present this dialog at startup' checkbox or by using the Options dialog under the Edit / Options menu. See also 2.14. Options.

Note: *The default password for the* `admin` *user is also* `admin`*. You should change this default password right after the creation using the Repository Explorer or the "Repository/Edit User" menu.*

3.8.1. Repository Auto-Login

You can have Spoon automatically log into the repository by setting the following environment variables: KETTLE_REPOSITORY, KETTLE_USER and KETTLE_PASSWORD.

This prevents you from having to log into the same repository every time.

Important Note: *this is a security risk and you should always lock your computer to prevent unauthorized access to the repository.*

3.9. License

Beginning with version 2.2.0, Kettle was released into the public domain under the LGPL license. Please refer to Appendix A for the full text of this license.

Note: *Pentaho Data Integration is referred to as "Kettle" below.*

```
Copyright (C) 2006 Pentaho Corporation

Kettle is free software; you can redistribute it and/or modify it under
the terms of the GNU Lesser General Public License as published by the
Free Software Foundation; either version 2.1 of the License, or (at your
option) any later version.

Kettle is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for
more details.

You should have received a copy of the GNU Lesser General Public License
along with the Kettle distribution; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
```

3.10. Definitions

3.10.1. Transformation Definitions

Value: Values are part of a row and can contain any type of data: Strings, floating point Numbers, unlimited precision BigNumbers, Integers, Dates or Boolean values.

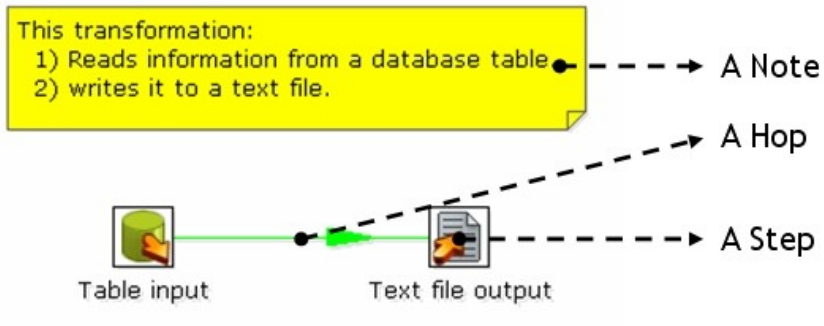
Row: a row exists of 0 or more values

Output stream: an output stream is a stack of rows that leaves a step.

Input stream: an input stream is a stack of rows that enters a step.

Hop: a hop is a graphical representation of one or more data streams between 2 steps. A hop always represents the output stream for one step and the input stream for another. The number of streams is equal to the copies of the destination step. (1 or more)

Note: a note is a descriptive piece of information that can be added to a transformation

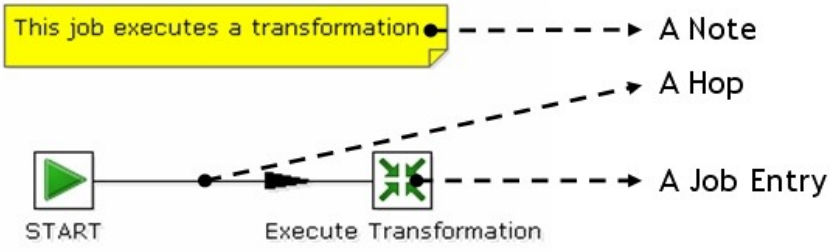


Job Definitions

Job Entry: A job entry is one part of a job and performs a certain task

Hop: a hop is a graphical representation of one or more data streams between 2 steps. A hop always represents the link between two job entries and can be set (depending on the type of originating job entry) to execute the next job entry unconditionally, after successful execution or failed execution.

Note: a note is a descriptive piece of information that can be added to a job



3.11. Toolbar

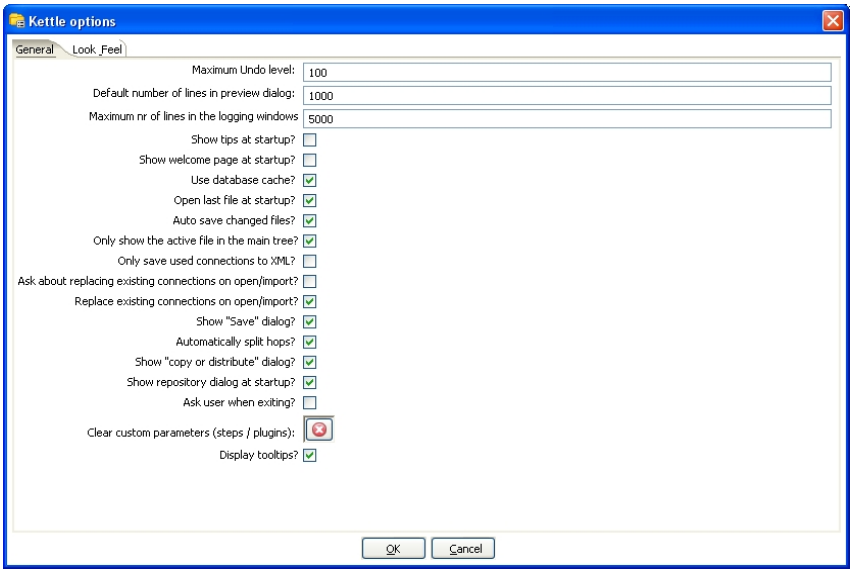
The icons on the toolbar of the main screen are from left to right:

Icon	Description
	Create a new job or transformation
	Open transformation/job from file if you're not connected to a repository or from the repository if you are connected to one.
	Save the transformation/job to a file or to the repository.
	Save the transformation/job under a different name or filename.
	Open the print dialog.
	Run transformation/job: runs the current transformation from XML file or repository.
	Preview transformation: runs the current transformation from memory. You can preview the rows that are produced by selected steps.
	Run the transformation in debug mode allowing you to troubleshoot execution errors.
	Replay the processing of a transformation for a certain date and time. This will cause certain steps (Text File Input and Excel Input) to only process rows that failed to be interpreted correctly during the run on that particular date and time.
	Verify transformation: Spoon runs a number of checks for every step to see if everything is going to run as it should.
	Run an impact analysis: what impact does the transformation have on the used databases.
	Generate the SQL that is needed to run the loaded transformation.
	Launches the database explorer allowing you to preview data, run SQL queries, generate DDL and more.

3.12. Options

Kettle options allow you to customize a number of properties related to the behavior and look and feel of the graphical user interface. Examples include startup options like whether or not to display tips and the Kettle Welcome Page, and user interface options like fonts and the colors. To access the options dialog, select Edit|Options... from the menubar.

3.12.1. General Tab



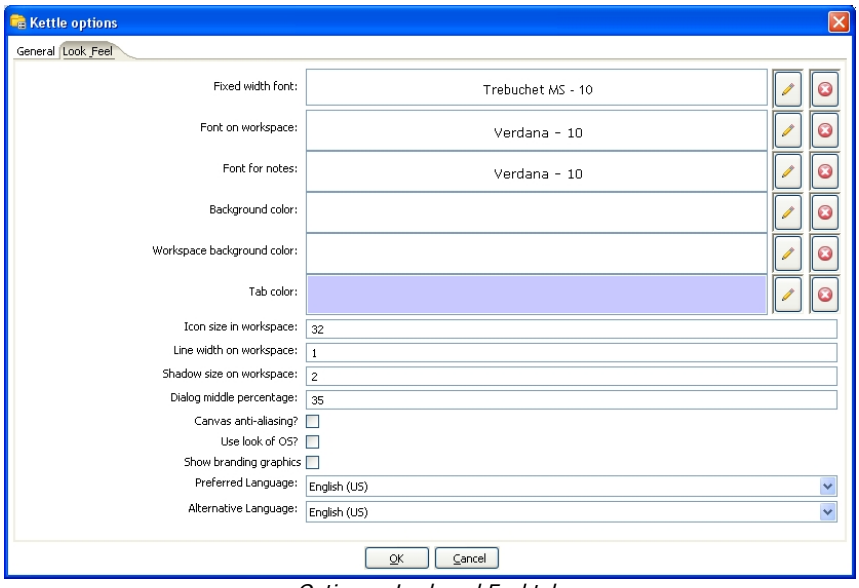
Options - General tab

Feature	Description
Maximum Undo Level	This parameter sets the maximum number of steps that can be undone (or redone) by Spoon.
Default number of lines in preview dialog	This parameter allows you to change the default number of rows that are requested from a step during transformation previews.
Maximum nr of lines in the logging windows	Specify the maximum limit of rows to display in the logging window.
Show tips at startup?	This options sets the display of tips at startup.
Show welcome page at startup?	This option controls whether or not to display the welcome page when launching Spoon.

Feature	Description
Use database cache?	Spoon caches information that is stored on source and target databases. In some cases this can lead to incorrect results when you're in the process of changing those very databases. In those cases it is possible to disable the cache altogether instead of clearing the cache every time. **NOTE:** *Spoon automatically clears the database cache when you launch DDL (Data Definition Language) statements towards a database connection. However, when using 3rd party tools, clearing the database cache manually may be necessary.*
Open last file at startup?	Enable this option to automatically (try to) load the last transformation you used (opened or saved) from XML or repository.
Auto save changed files?	This option automatically saves a changed transformation before running.
Only show the active file in the main tree?	This option reduces the number of transformation and job items in the main tree on the left by only showing the currently active file.
Only save used connections to XML?	This option limits the XML export of a transformation to the used connections in that transformation. This comes in handy while exchanging sample transformations to avoid having all defined connections to be included.
Ask about replacing existing connections on open/import?	This option asks before replacing existing database connections during import.
Replace existing connections on open/import?	This is the action that's being taken when there is no dialog shown. (see previous option)
Show "Save" dialog?	This flag allows you to turn off the confirmation dialogs you receive when a transformation has been changed.
Automatically split hops?	This option turns off the confirmation dialogs you get when you want to split a hop. (see also 7.4. Splitting A Hop)
Show "copy or distribute" dialog?	This options turns off the warning message that appears when you link a step to multiple outputs. This warning message describes the two options for handling multiple outputs: Distribute rows – destination steps receive the rows in turns (round robin) Copy rows – all rows are sent to all destinations
Show repository dialog at startup?	This option controls whether or not the repositories dialog shows up at startup.
Ask user when exiting?	This option controls whether or not to display the confirmation dialog when a user chooses to exit the application.
Clear custom parameters (steps/plugins)	This option clears all parameters and flags that were set in the plugin or step dialogs.

Feature	Description
Display tootlips?	This option controls whether or not to display tooltips for the buttons on the main toolbar.

3.12.2. Look & Feel tab

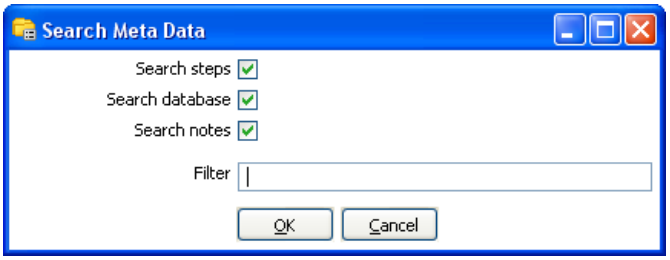


Options – Look and Feel tab

Feature	Description
Fixed width font	This is the font that is used in the dialog boxes, trees, input fields, etc.
Font on workspace	This is the font that is used on the graphical view.
Font for notes	This font is used in the notes that are displayed in the Graphical View.
Background color	Sets the background color in Spoon. It affects all dialogs too.
Workspace background color	Sets the background color in the Graphical View of Spoon.
Tab color	This is the color that is being used to indicate tabs that are active/selected.
Icon size in workspace	This affects the size of the icons in the graph window. The original size of an icon is 32x32 pixels. The best results (graphically) are probably at sizes 16,24,32,48,64 and other multiples of 32.
Line width on workspace	This affects the line width of the hops on the Graphical View and the border around the steps.
Shadow size on workspace	If this size is larger then 0, a shadow of the steps, hops and notes is drawn on the canvas, making it look like the transformation floats

Feature	Description
	above the canvas.
Dialog middle percentage	By default, a parameter is drawn at 35% of the width of the dialog, counted from the left. You can change this with this parameter. Perhaps this can be useful in cases where you use unusually large fonts.
Canvas anti-aliasing?	Some platforms like Windows, OSX and Linux support anti-aliasing through GDI, Carbon or Cairo. Check this to enable smoother lines and icons in your graph view. If you enable this and your environment doesn't work any more afterwards, change the value for option "EnableAntiAliasing" to "N" in file $HOME/.kettle/.spoonrc (C:\Documents and Settings\<user>\.kettle\.spoonrc on Windows)
Use look of OS?	Checking this on Windows allows you to use the default system settings for fonts and colors in Spoon. On other platforms, this is always the case.
Show branding graphics	Enabling this option will draw Pentaho Data Integration branding graphics on the canvas and in the left hand side "expand bar".
Preferred Language	Here you can specify the default language setting. If a certain text hasn't been translated into this locale, Kettle will fall back to the fail over locale.
Alternative Language	Because the original language in which Kettle was written is English, it's best to set this locale to English.

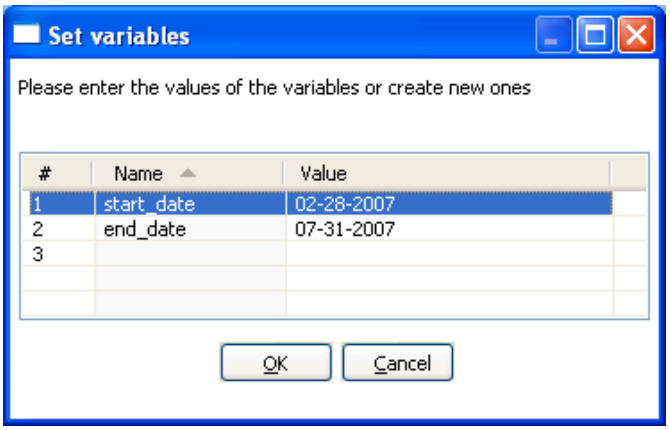
3.13. Search Meta data



Search Meta data Dialog

This option will search in any available fields, connectors or notes of all loaded jobs and transformations for the string specified in the Filter field. The Meta data search returns a detailed result set showing the location of any search hits. This feature is accessed by choosing Edit|Search Meta data from the menubar.

3.14. Set environment variable



Set Environment Variable Dialog

The Set Environment Variable feature allows you to explicitly create and set environment variables for the current user session. This is a useful feature when designing transformations for testing variable substitutions that are normally set dynamically by another job or transformation.

This feature is accessible by choosing Edit|Set Environment Variable from the menubar.

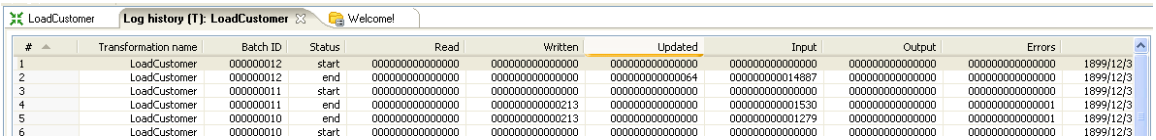
Note: This screen is also presented when you run a transformation that use undefined variables. This allows you to define them right before execution time.

Show environment variables

This feature will display the current list of environment variables and their values. It is accessed by selecting the Edit|Show environment variables option from the menubar.

3.15. Execution log history

If you have configured your Job or Transformation to store log information in a database table, you can view the log information from previous executions by right-clicking on the job or transformation in the Main Tree and selecting 'Open History View'. This view will show



#	Transformation name	Batch ID	Status	Read	Written	Updated	Input	Output	Errors	
1	LoadCustomer	000000012	start	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	1899/12/3
2	LoadCustomer	000000012	end	000000000000000	000000000000000	000000000000064	000000000014887	000000000000000	000000000000000	1899/12/3
3	LoadCustomer	000000011	start	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	1899/12/3
4	LoadCustomer	000000011	end	000000000000000	000000000000213	000000000000000	000000000001530	000000000000000	000000000000001	1899/12/3
5	LoadCustomer	000000010	end	000000000000000	000000000000213	000000000000000	000000000001279	000000000000000	000000000000001	1899/12/3
6	LoadCustomer	000000010	start	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	000000000000000	1899/12/3

Transformation History Tab

NOTE: *The log history for a job or transformation will also open by default each next time you execute the file.*

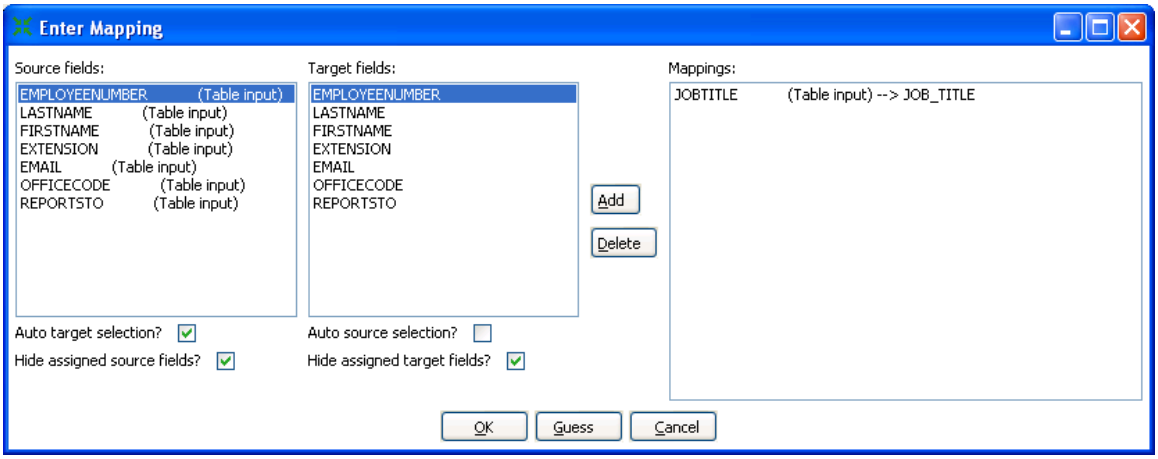
3.16. Replay

The Replay feature allows you to re-run a transformation that failed. Replay functionality is implemented for Text File Input and Excel input. It allows you to send files that had errors back to the source and have the data corrected. ONLY the lines that failed before are then processed during the replay if a .line file is present. It uses the date in the filename of the .line file to match the entered replay date.

3.17. Generate mapping against target step

In cases where you have a fixed target table, you will want to map the fields from the stream to their corresponding fields in the target output table. This is normally accomplished using a Select Values step in your transformation. The 'Generate mapping against target' option provides you with an easy-to-use dialog for defining these mappings that will automatically create the resulting Select Values step that can be dropped into your transformation flow prior to the table output step.

To access the 'Generate mapping against target' option is accessed by right-clicking on the table output step.



Generate Mapping Dialog

After defining your mappings, select OK and the Select Values step containing your mappings will appear on the workspace. Simply, attach the mapping step into your transformation immediatelyAttach the mapping step into your transformation just before the table output step.

3.17.1. Generate mappings example

Here is an example of a simple transformation in which we want to generate mappings to our target output table:

Split hop before generating mappings

Begin by right-clicking on the Table output step and selecting 'Generate mappings against target'. Add all necessary mappings using the Generate Mapping dialog shown above and click OK. You will now see a Table output mapping step has been added to the canvas:

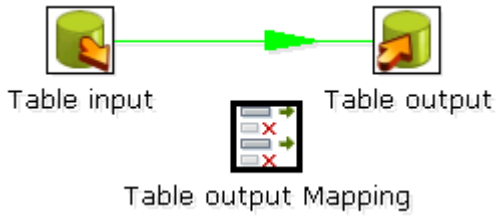
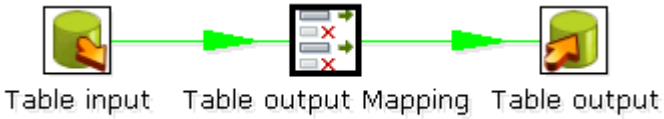


Table output Mapping Step added to canvas

Finally, drag the generated Table output Mapping step into your transformation flow prior to the table output step:



Insert mapping step into transformation flow

3.18. Safe mode

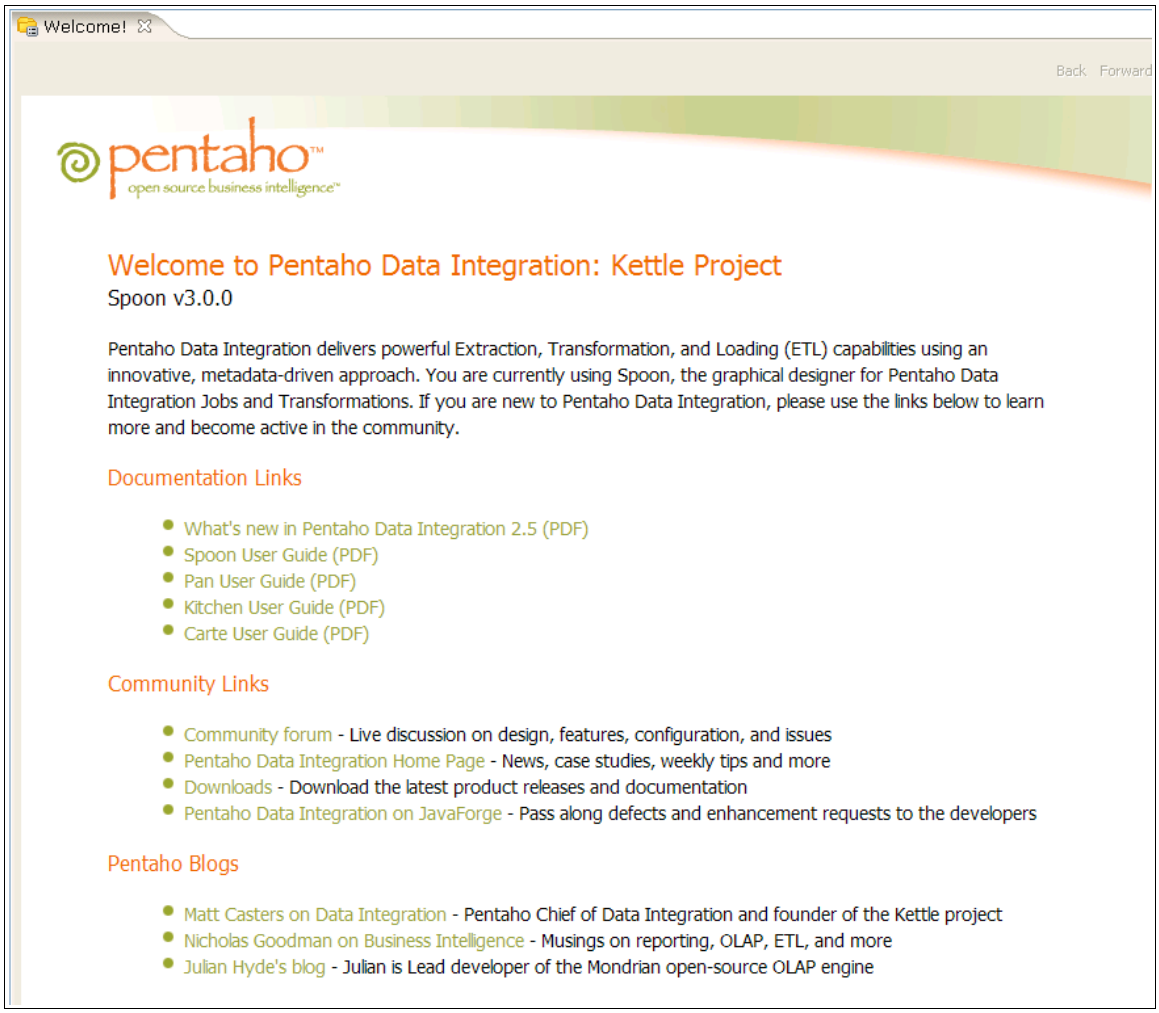
In cases where you are mixing the rows from various sources, you need to make sure that these row all have the same layout in all conditions. For this purpose, we added a "safe mode" option that is available in the Spoon logging window or on the Execute a Transformation/Job window. When running in "safe mode", the transformation will check every row that passes and will see if the layouts are all identical.

If a row is found that does not have the same layout as the first row, an error is thrown and the step and offending row are reported on.

Note: *this option is also available in Pan with the "safe mode" option.*

3.19. Welcome Screen

The welcome screen will display the first time you launch Spoon 3.0 providing you with links to additional information about Pentaho Data Integration. You can disable the launching of the welcome page in Spoon options (Edit|Options).

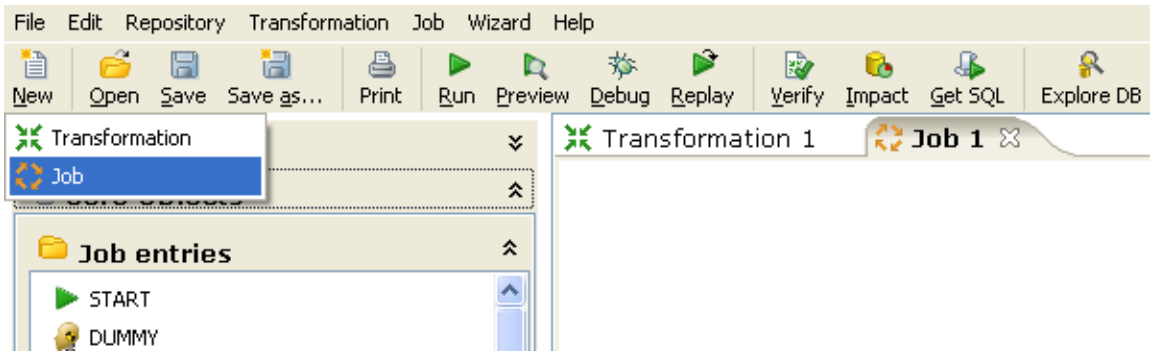


The welcome screen

4. Creating a Transformation or Job

You create a new Transformation by clicking on the New Transformation button on the main toolbar, by selecting File|New|Transformation from the menubar, or by using the CTRL-N hotkey. This will open a new Transformation tab for you to being designing your transformation.

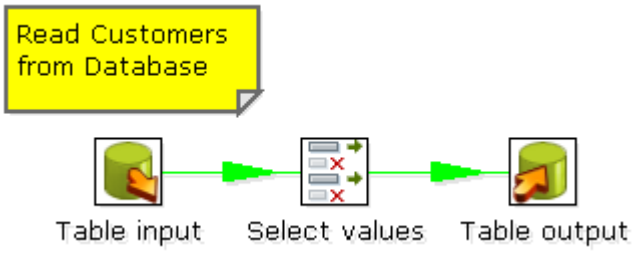
You create a new Job by clicking on the New Job button on the main toolbar, by selecting File|New|Job from the menubar, or by using the CTRL-ALT-N hotkey. This will open an new Job tab for you to begin designing your job.



Creating a new transformation or job

4.1. Notes

Notes allow you to add descriptive text notes to the Job or Transformation canvas. To add a note to the graphical view, right-click on the canvas and select 'Add note'. Later, these notes can be edited by double clicking on them and dragged around the screen by dragging on them with the mouse using the left button. To remove a note, right-click on the note and select 'Delete note'.

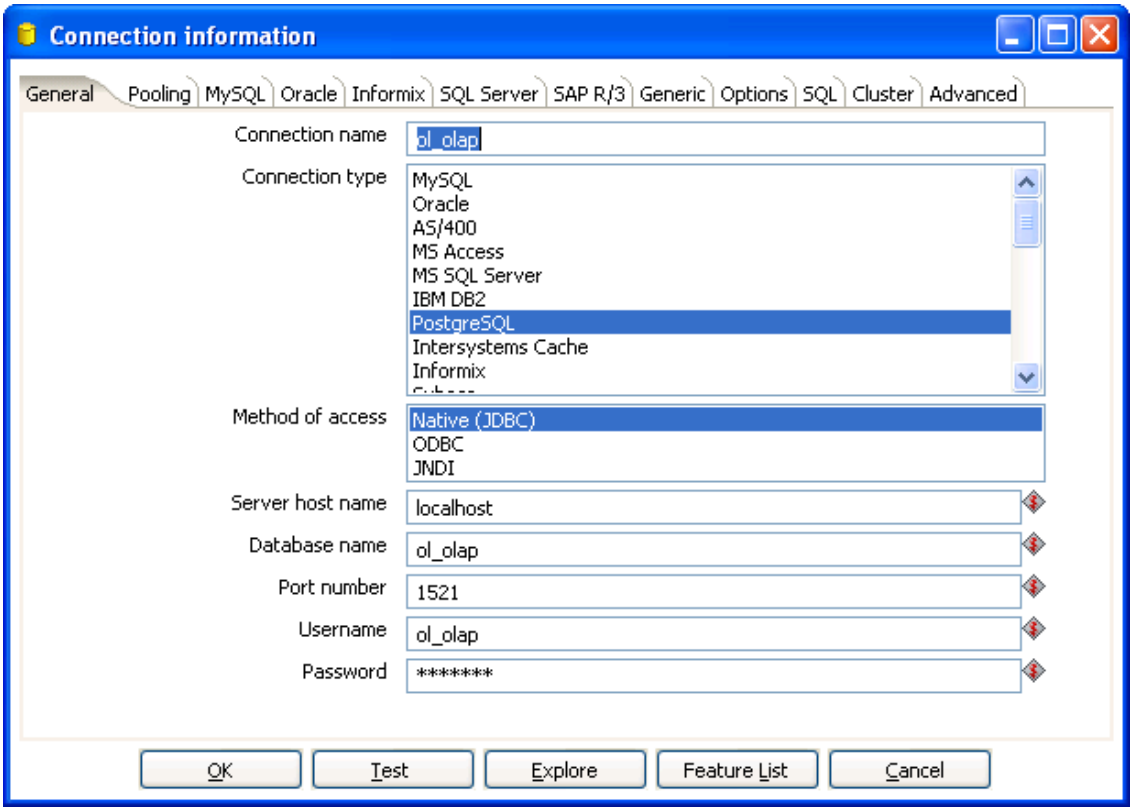


Notes

5. Database Connections

A database connection describes the method by which Kettle can connect to a database. You can create connections specific to a Job or Transformation or store them in the Kettle repository for re-use within multiple transformations or jobs.

5.1. Screen shot

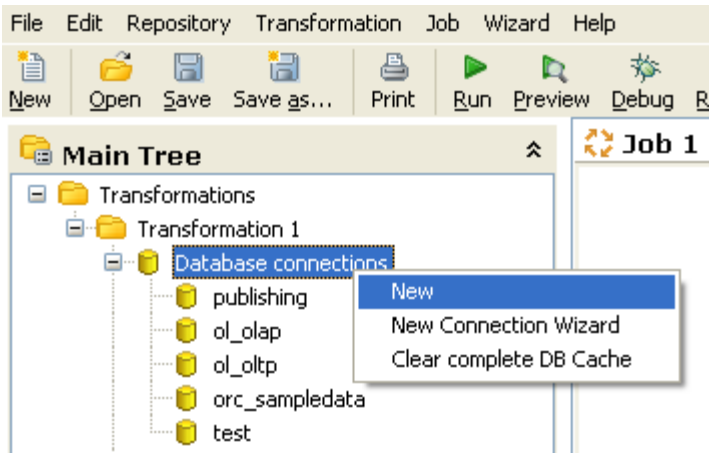


The Connection information dialog

5.2. Creating a new database connection

This section describes how to create a new database connection including a detailed description of each connection property available in the Connection information dialog.

You begin creating a new connection by right-clicking on the 'Database Connections' tree entry and selecting 'New' or 'New Connection Wizard', by double-clicking on 'Database Connections', or simply by pressing F3.



Creating a new database connection

This will launch the 'Connection information' dialog shown above. The following topics describe the configuration options available on each tab of the Connection information dialog.

5.2.1. General

The general tab is where you setup the basic information about your connection like the connection name, type, access method, server name and login credentials. The table below provides a more detailed description of the options available on the General tab:

Feature	Description
Connection Name	Uniquely identifies a connection across transformations and jobs
Connection Type	The type of database you are connecting to (i.e. MySQL, Oracle, etc.)
Method of access	This will be either Native (JDBC), ODBC, or OCI. Available access types are dependent on the type of database you are connecting to
Server host name	Defines the host name of the server on which the database resides. You can also specify the host by IP-address
Database name	Identifies the database name you want to connect to. In case of ODBC, specify the DSN name here
Port number	Sets the TCP/IP port number on which the database listens
Username	Optionally specifies the username to connect to the database
Password	Optionally specifies the password to connect to the database

5.2.2. Pooling

The pooling tab allows you to configure your connection to use connection pooling and define options related to connection pooling like the initial pool size, maximum pool size and connection pool parameters. The table below provides a more detailed description of the options available on the Pooling tab:

Feature	Description
Use a connection pool	Check this option to enable connection pooling.
The initial pool size	Sets the initial size of the connection pool.
The maximum pool size.	Sets the maximum number of connections in the connection pool.
Parameter Table	Allows you to define additional custom pool parameters.

5.2.3. MySQL

Because by default, MySQL gives back complete query results in one block to the client (Kettle in this case) we had to enable "result streaming" by default. The big drawback of this is that it allows only 1 (one) single query to be opened at any given time. If you run into trouble because of that, you can disable this option in the MySQL tab of the database connection dialog.

Another issue you might come across is that the default timeout in the MySQL JDBC driver is set to 0. (no timeout) This leads to a problem in certain situations as it doesn't allow Kettle to detect a server crash or sudden network failure if it happens in the middle of a query or open database connection. This in turn leads to the infinite stalling of a transformation or job. To solve this, set the "connectTimeout" and "socketTimeout" parameters for MySQL in the Options tab. The value to be specified is in milliseconds: for a 2 minute timeout you would specify value 120000 (2 x 60 x 1000).

You can also review other options on the linked MySQL help page by clicking on the 'Show help text on option usage' button found on the Options tab.

5.2.4. Oracle

This tab allows you to specify the default data and index tablespaces which Kettle will use when generating SQL for Oracle tables and indexes.

This version of Pentaho Data Integration ships with the Oracle JDBC driver version 10.2.0. It is in general the most stable and recent driver we could find. However, if you do have issues with Oracle connectivity or other strange problems, you might want to consider replacing the 10.2. JDBC driver to match your database server. Replace files "ojdbc14.jar" and "orai18n.jar" in the directory libext/JDBC of your distribution with the files found in the $ORACLE_HOME/jdbc directory on your server.

If you want to use OCI and an Oracle Net8 client, please read on. For OCI to work, the JDBC driver version used in Kettle needs to match your Oracle client version. Oracle 2.5.0 shipped with version 10.1, 2.5.0 ships with version 10.2.
You can either install that version of the Oracle client or (probably easier) change the JDBC driver in PDI if versions don't match up. (see above)

5.2.5. Informix

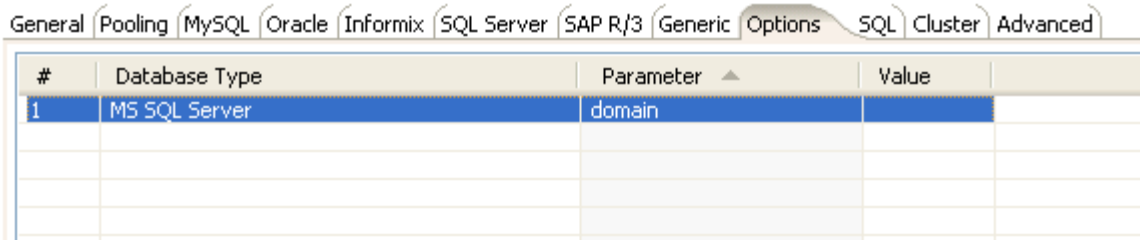
For Informix, you need to specify the Informix Server name in the Informix tab in order for a connection to be usable.

5.2.6. SQL Server

This tab allows you configure the following properties specific to Microsoft SQL Server:

Feature	Description
SQL Server instance name	Sets the instance name property for the SQL Server connection.
Use .. to separate schema and table	Enable when using dot notation to separate schema and table.

Other properties can be configured by adding connection parameters on the options tab of the Connection information dialog. For example, you can enable single sign-on login by defining the ***domain*** option on the Options tab as shown below:



The SQL Server "instance" property

From the jTDS FAQ on http://jtds.sourceforge.net/faq.html:

> *Specifies the Windows domain to authenticate in. If present and the user name and password are provided, jTDS uses Windows (NTLM) authentication instead of the usual SQL Server authentication (i.e. the user and password provided are the domain user and password). This allows non-Windows clients to log in to servers which are only configured to accept Windows authentication.*
>
> *If the domain parameter is present but no user name and password are provided, jTDS uses its native Single-Sign-On library and logs in with the logged Windows user's credentials (for this to work one would obviously need to be on Windows, logged into a domain, and also have the SSO library installed -- consult README.SSO in the distribution on how to do this).*

5.2.7. SAP R/3

This tab allows you configure the following properties specific to SAP R/3:

Feature	Description
Language	Specifies the language to be used when connecting to SAP.
System Number	Specifies the system number of the SAP system to which you want to connect.
SAP Client	Specifies the three digit client number for the connection.

5.2.8. Netezza

To avoid version compatibility issues, Pentaho Data Integration does not bundle the Netezza JDBC driver. If you do not have the JDBC driver, you can request a copy by contacting Netezza customer support (https://support.netezza.com).

To configure the Netezza driver with Pentaho Data Integration, simply copy the driver file (nzjdbc.jar) into your .\KETTLE-HOME\libext\JDBC directory.

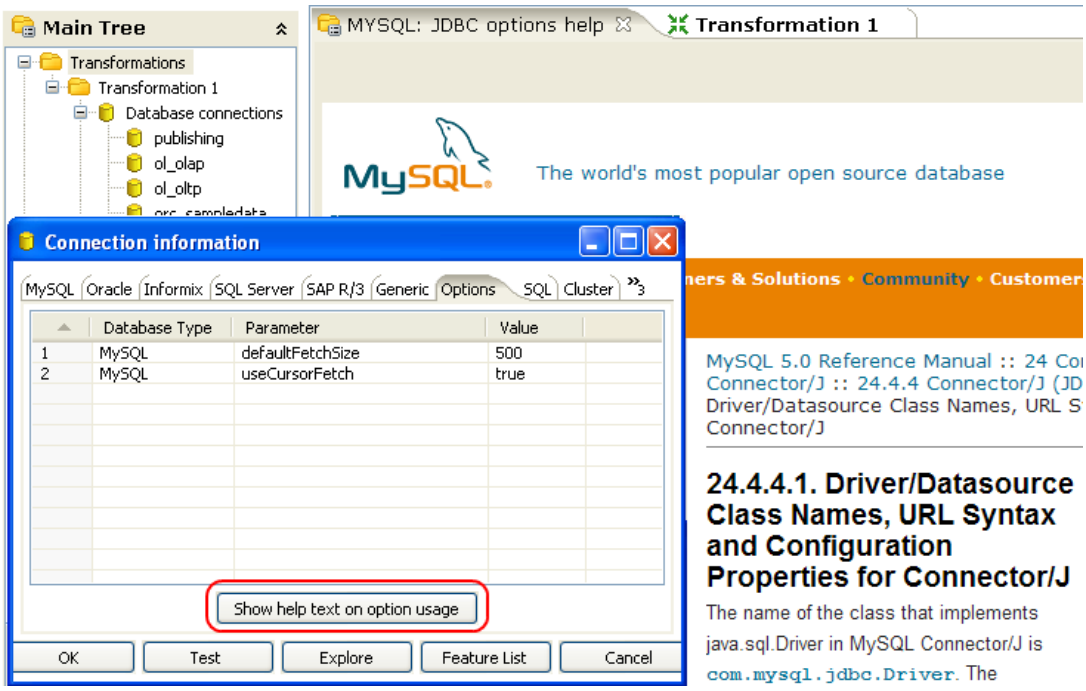
5.2.9. Generic

This tab is where you specify the URL and Driver class for Generic Database connections! You can also dynamically set these properties using Kettle variables. This provides the ability to access data from multiple database types using the same transformations and jobs.

Note: *Make sure to use clean ANSI SQL that works on all used database types in that case.*

5.2.10. Options

This tab allows you to set database specific options for the connection by adding parameters to the generated URL. To add a parameter, select the next available row in the parameter table, choose your database type, then enter a valid parameter name and its corresponding value. For more database specific configuration help, click the 'Show help text on option usage' button and a new browser tab will appear in Spoon with additional information about the configuring the JDBC connection for the currently selected database type:



Display options help in a Spoon browser

5.2.11. SQL

This tab allows you to enter a number of SQL commands immediately after connecting to the database. This is sometimes needed for various reasons like licensing, configuration, logging, tracing, etc.

5.2.12. Cluster

This tab allows you to enable clustering for the database connection and create connections to the data partitions. To enable clustering for the connection, check the 'Use Clustering?' option.

To create a new data partition, enter a partition ID and the hostname, port, database, username and password for connecting to the partition.

5.2.13. Advanced

This tab allows you configure the following properties for the connection:

Feature	Description
Quote all identifiers in database	Put quotes around all identifiers in SQL statements, without this quotes will be put around identifiers when it's needed only.
Force all identifiers to lower case	Force all identifiers to lower case.
Force all identifiers to upper case	Force all identifiers to upper case.

5.2.14. Test a connection

The 'Test' button in the Connection information dialog allows you to test the current connection. An OK message will be displayed if Spoon is able to establish a connection with the target database.

5.2.15. Explore

The Database Explorer allows you to interactively browse the target database, preview data, generate DDL and much more. To open the Database Explorer for an existing connection, click the 'Explore' button found on the Connection information dialog or right-click on the connection in the Main tree and select 'Explore'. Please see Database Explorer for more information.

5.2.16. Feature List

Feature list: exposes the JDBC URL, class and various database settings for the connection such as the list of reserved words.

5.3. Editing a connection

To edit an existing connection, double-click on the connection name in the main tree or right-click on the connection name and select "Edit connection".

5.4. Duplicate a connection

To duplicate an existing connection, right-click on the connection name and select "Duplicate".

5.5. Copy to clipboard

Accessed by right-clicking on a connection name in the main tree, this option copies the XML describing the connection to the clipboard.

5.6. Delete a connection

To delete an existing database connection, right-click on the connection name in the main tree and select "Delete".

5.7. Execute SQL commands on a connection

To execute SQL command against an existing connection, right-click on the connection name and select "SQL Editor". See also SQL Editor for more information.

5.8. Clear DB Cache option

To speed up connections Spoon uses a database cache. When the information in the cache no longer represents the layout of the database, right-click on the connection in the Main tree and select the 'Clear DB Cache...' option. This is commonly used when databases tables have been changed, created or deleted.

5.9. Quoting

We had more and more people complain about the handling of reserved words, field names with spaces in it, field names with decimals (.) in it, table names with dashes and other special characters in it ... we implemented a database specific quoting system that allows you to pretty much use any name or character that the database is comfortable with.

Pentaho Data Integration contains a list of reserved words for many (but not all) of the supported databases. To correctly implement quoting, we had to go for a strict separation between the schema (user/owner) of a table and the tablename itself. Otherwise it would be impossible to properly quote tables or fields with one or more periods in them. Putting dots in table and field names is apparently common practice in certain ERP systems. (for example fields like "V.A.T.")

Because we too can be wrong when doing the quoting, we have added a new rule in version 2.5.0: when there is a start or end-quote in the tablename or schema, Pentaho Data Integration refrains from doing the quoting. This allows you to specify the quoting mechanism yourself. This leaves you all the freedom you need to get out of any sticky situation that might be left. Nevertheless, feel free to let us know about it so that we can improve our quoting algorithms.

5.10. Database Usage Grid

Database	Access Method	Server Name or IP Address	Database Name	Port # (default)	Username & Password
Oracle	Native	Required	Oracle database SID	Required (1521)	Required
	ODBC		ODBC DSN name		Required
	OCI		Database TNS name		Required
MySQL	Native	Required	MySQL database name	Optional (3306)	Optional
	ODBC		ODBC DSN name		Optional
AS/400	Native	Required	AS/400 Library name	Optional	Required
	ODBC		ODBC DSN name		Required
MS Access	ODBC		ODBC DSN name		Optional
MS SQL Server	Native	Required	Database name	Required (1433)	Required
	ODBC		ODBC DSN name		Required
IBM DB2	Native	Required	Database name	Required (50000)	Required

Database	Access Method	Server Name or IP Address	Database Name	Port # (default)	Username & Password
	ODBC		ODBC DSN name		Required
PostgreSQL	Native	Required	Database name	Required (5432)	Required
	ODBC		ODBC DSN name		Required
Intersystems Caché	Native	Required	Database name	Required (1972)	Required
	ODBC		ODBC DSN name		Required
Sybase	Native	Required	Database name	Required(5001)	Required
	ODBC		ODBC DSN name		Required
Gupta SQL Base	Native	Required	Database Name	Required (2155)	Required
	ODBC		ODBC DSN name		Required
Dbase III,IV or 5.0	ODBC		ODBC DSN name		Optional
Firebird SQL	Native	Required	Database name	Required (3050)	Required
	ODBC		ODBC DSN name		Required
Hypersonic	Native	Required	Database name	Required (9001)	Required
MaxDB (SAP DB)	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Ingres	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Borland Interbase	Native	Required	Database name	Required (3050)	Required
	ODBC		ODBC DSN name		Required
ExtenDB	Native	Required	Database name	Required (6453)	Required
	ODBC		ODBC DSN name		Required
Teradata	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Oracle RDB	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
H2	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Netezza	Native	Required	Database name	Required (5480)	Required

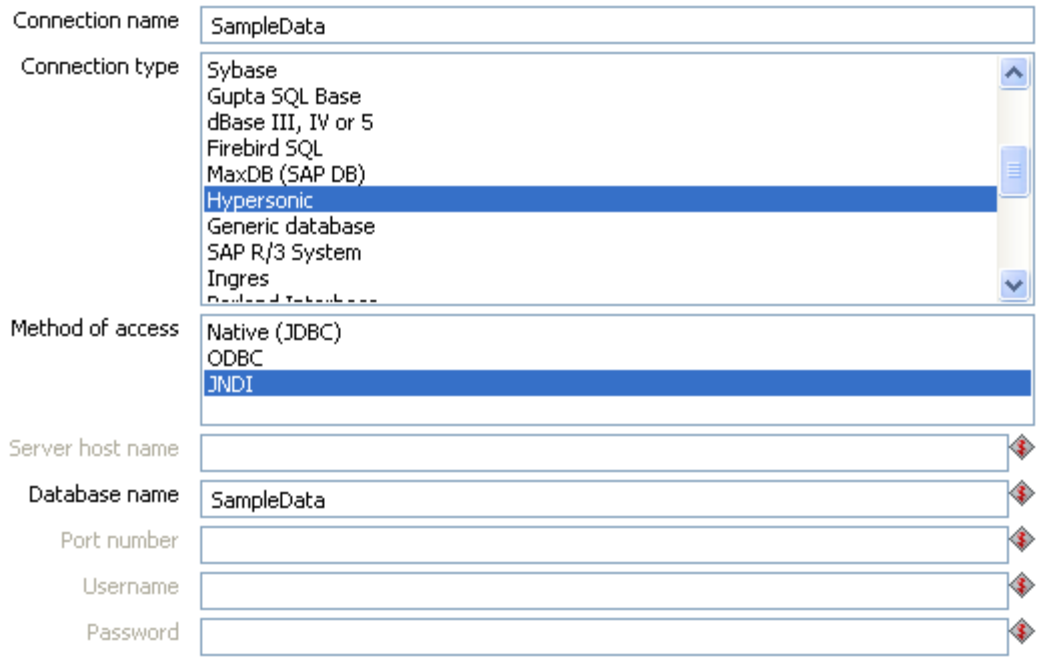
Database	Access Method	Server Name or IP Address	Database Name	Port # (default)	Username & Password
	ODBC		ODBC DSN name		Required
IBM Universe	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
SQLite	Native	Required	Database name		Required
	ODBC		ODBC DSN name		Required
Apache Derby	Native	optional	Database name	Optional (1527)	Optional
	ODBC		ODBC DSN name		Optional
Generic (*)	Native	Required	Database name	Required (Any)	Required
	ODBC		ODBC DSN name		Optional

(*) The generic database connection also needs to specify the URL and Driver class in the Generic tab! We now also allow these fields to be specified using a variable. That way you can access data from multiple database types using the same transformations and jobs. Make sure to use clean ANSI SQL that works on all used database types in that case.

5.11. Configuring JNDI connections

If you are developing transformations and jobs that will be deployed on an application server such as the Pentaho platform running on JBoss, you can configure your database connections using JNDI.
Because you don't want to have an application server running all the time during development or testing of the transformations, we have supplied a way of configuring a JNDI connection for "local" Kettle use.
To configure, edit properties file called "simple-jndi/jdbc.properties"
For example, to connect to the databases used in Pentaho Demo platform download, use this information in the properties file:

```
SampleData/type=javax.sql.DataSource
SampleData/driver=org.hsqldb.jdbcDriver
SampleData/url=jdbc:hsqldb:hsql://localhost/sampledata
SampleData/user=pentaho_user
SampleData/password=password
Quartz/type=javax.sql.DataSource
Quartz/driver=org.hsqldb.jdbcDriver
Quartz/url=jdbc:hsqldb:hsql://localhost/quartz
Quartz/user=pentaho_user
Quartz/password=password
Hibernate/type=javax.sql.DataSource
Hibernate/driver=org.hsqldb.jdbcDriver
Hibernate/url=jdbc:hsqldb:hsql://localhost/hibernate
Hibernate/user=hibuser
Hibernate/password=password
Shark/type=javax.sql.DataSource
Shark/driver=org.hsqldb.jdbcDriver
Shark/url=jdbc:hsqldb:hsql://localhost/shark
Shark/user=sa
Shark/password=
```



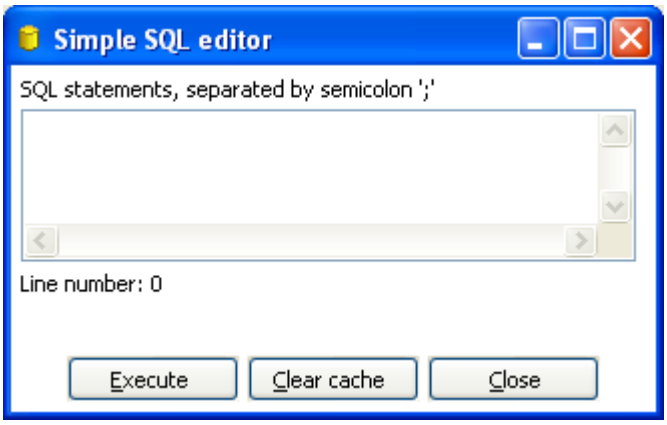
Note: *It is important that the information stored in this file in the simple-jndi directory mirrors the content of your application server data sources.*

5.12. Unsupported databases

If you want to access a database type that is not yet supported, let us know and we will try to find a solution. A few database types are not supported in this release because of the lack of sample database and/or software.

Please note that it is usually still possible to read from these databases by using the Generic database driver through an ODBC or JDBC connection.

6. SQL Editor



The Simple SQL Editor dialog

6.1. Description

The Simple SQL Editor is an easy-to-use tool when you need to execute standard SQL commands for tasks like creating tables, dropping indexes and modifying fields. In several places throughout Spoon, the SQL Editor is used to preview and execute DDL (Data Definition Language) generated by Spoon such as "create/alter table", "create index" and "create sequence" SQL commands. An example of this would be if you added at Table Output step to a transformation and clicked the SQL button at the bottom of the Table Input dialog. Spoon will automatically generate the necessary DDL for the output step to function properly and present that to the end user via the SQL Editor.

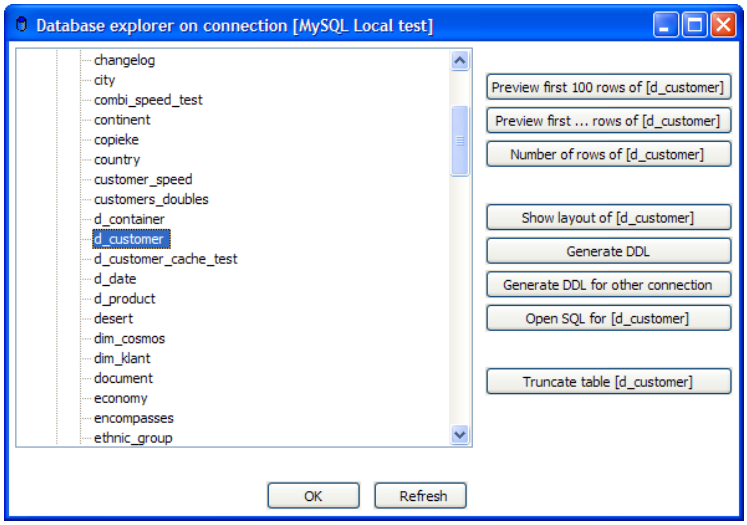
Notes:

- *Multiple SQL Statements have to be separated by semi-colons (;).*
- *Before these SQL Statements are sent to the database to be executed, Spoon removes returns, line-feeds and the separating semi-colons.*
- *Kettle clears the database cache for the database connection on which you launch DDL statements.*

6.2. Limitations

This is a simple SQL editor. It does not know all the dialects of all the more than 20 supported databases. That means that creating stored procedures, triggers and other database specific objects might pose problems. Please consider using the tools that came with the database in that case.

7. Database Explorer



The database explorer dialog

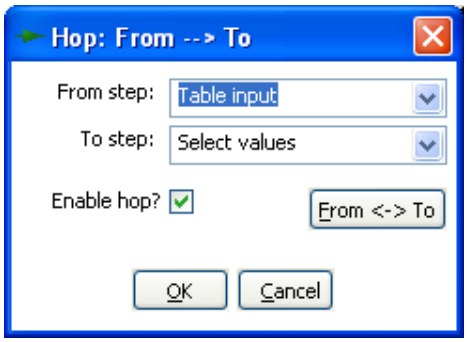
7.1. Description

The Database Explorer provides the ability to explore configured database connections. It currently supports tables, views and synonyms along with the catalog and/or schema to which the table belongs.

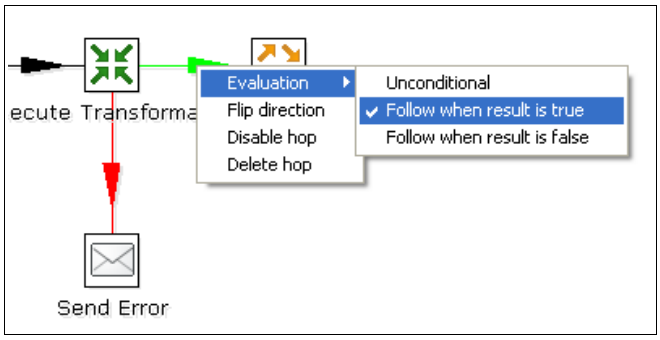
The buttons to the right provide quick access the following features for the selected table:

Feature	Description
Preview first 100 rows of...	Returns the first 100 rows from the selected table
Preview first ... rows of...	Prompts the user for the number of rows to return from the selected table
Number of rows...	Count the number of rows in the selected table
Show Layout of...	Displays a list of column names, data types, etc. from the selected table
Generate DDL	Generates the DDL to create the selected table based on the current connection type
Generate DDL for other connection	Prompts the user for another connection, then generates the DDL to create the selected table based on the user selected connection type.
Open SQL for...	Launches the Simple SQL Editor for the selected table
Truncate table...	Generates a TRUNCATE table statement for the current table. **Note:** *The statement is commented out by default to prevent the user from accidentally deleting the table data.*

8. Hops



Editing a Transformation Hop



Editing a Job Hop

8.1. Description

A hop connects one transformation step or job entry with another. The direction of the data flow is indicated with an arrow on the graphical view pane. A hop can be enabled or disabled (for testing purposes for example).

8.1.1. Transformation Hops

When a hop is disabled in a transformation, the steps downstream of the disabled hop are cut off from any data flowing upstream of the disabled hop. This may lead to unexpected results when editing the downstream steps. For example, if a particular step-type offers a "Get Fields" button, clicking the button may not reveal any of the incoming fields as long as the hop is still disabled.

8.1.2. Job Hops

Besides the execution order, it also specifies the condition on which the next job entry will be executed. You can specify the evaluation mode by right clicking on the job hop:

- "Unconditional" specifies that the next job entry will be executed regardless of the result of the originating job entry.
- "Follow when result is true" specifies that the next job entry will only be executed when the result of the originating job entry was true, meaning successful execution, file found, table found, without error, evaluation was true, ...
- "Follow when result is false" specifies that the next job entry will only be executed when the result of the originating job entry was false, meaning unsuccessful execution, file not found, table not found, error(s) occurred, evaluation was false, ...

8.2. Creating A Hop

You can easily create a new hop between 2 steps by one of the following options:

- Dragging on the Graphical View between 2 steps while using the middle mouse button.
- Dragging on the Graphical View between 2 steps while pressing the SHIFT key and using the left mouse button.

- Selecting two steps in the tree, clicking right and selecting "new hop"
- Selecting two steps in the graphical view (CTRL + left mouse click), right clicking on a step and selecting "new hop"
- Splitting A Hop

You can easily insert a new step into a new hop between two steps by dragging the step (in the Graphical View) over a hop until the hop becomes drawn in bold. Release the left button and you will be asked if you want to split the hop. This works only with steps that have not yet been connected to another step.

8.3. Loops

Loops are not allowed in transformations because Spoon depends heavily on the previous steps to determine the field values that are passed from one step to another. If we would allow loops in transformations we often would get endless loops and undetermined results.

Loops **are** allowed in jobs because Spoon executes job entries sequentially. Just make sure you don't build endless loops. This job entry can help you exit closed loops based on the number of times a job entry was executed.

8.4. Mixing rows: trap detector

Mixing rows with different layout is not allowed in a transformation. Mixing row layouts will cause steps to fail because fields can not be found where expected or the data type changes unexpectedly.

The "trap detector" is in place to provide warnings at design time when a step is receiving mixed layouts:

In this case, the full error report reads:

> *We detected rows with varying number of fields, this is not allowed in a transformation. The first row contained 13 fields, another one contained 16 : [customer_tk=0, version=0, date_from=, date_to=, CUSTOMERNR=0, NAME=, FIRSTNAME=, LANGUAGE=, GENDER=, STREET=, HOUSNR=, BUSNR=, ZIPCODE=, LOCATION=, COUNTRY=, DATE_OF_BIRTH=]*

Note: *this is only a warning and will not prevent you from performing the task you want to do.*

8.5. Transformation hop colors

Transformation hops display in a variety of colors based on the properties and state of the hop. The following table describes the meaning behind a transformation hop's color:

Hop Color	Meaning
Green	Distribute rows: if multiple hops are leaving a step, rows of data will be evenly distributed to all target steps.
Red	Copies rows: if multiple hops are leaving a step, all rows of data will be copied to all target steps.
Yellow	Provides info for step, distributes rows
Magenta	Provides info for step, copies rows
Gray	The hop is disabled.
Black	The hop has a named target step.
Blue	Candidate hop using middle button + drag
Orange (Dot line)	The hop is never used because no data will ever go there.
Red (Bold Dot line)	The hop is used for carrying rows that caused errors in source step(s).

9. Variables

9.1. Variable usage

Variables can be used throughout Pentaho Data Integration, including within transformation steps and job entries. Variables can be defined by setting them with the Set Variable step in a transformation or by setting them in the `kettle.properties` file in the directory:

```
$HOME/.kettle (Unix/Linux/OSX)
C:\Documents and Settings\<username>\.kettle\ (Windows)
```

The way to use them is either by grabbing them using the Get Variable step or by specifying meta-data strings like:

```
${VARIABLE} or
%%VARIABLE%%
```

Both formats can be used and even mixed, the first is a UNIX derivative, the second is derived from Microsoft Windows. Dialogs that support variable usage throughout Pentaho Data Integration are visually indicated using a red dollar sign like this:

You can use CTRL-SPACE hotkey to select a variable to be inserted into the property value. Mouse over the variable icon to see a shortcut help text displayed.

9.2. Variable scope

The scope of a variable is defined by the place in which it is defined.

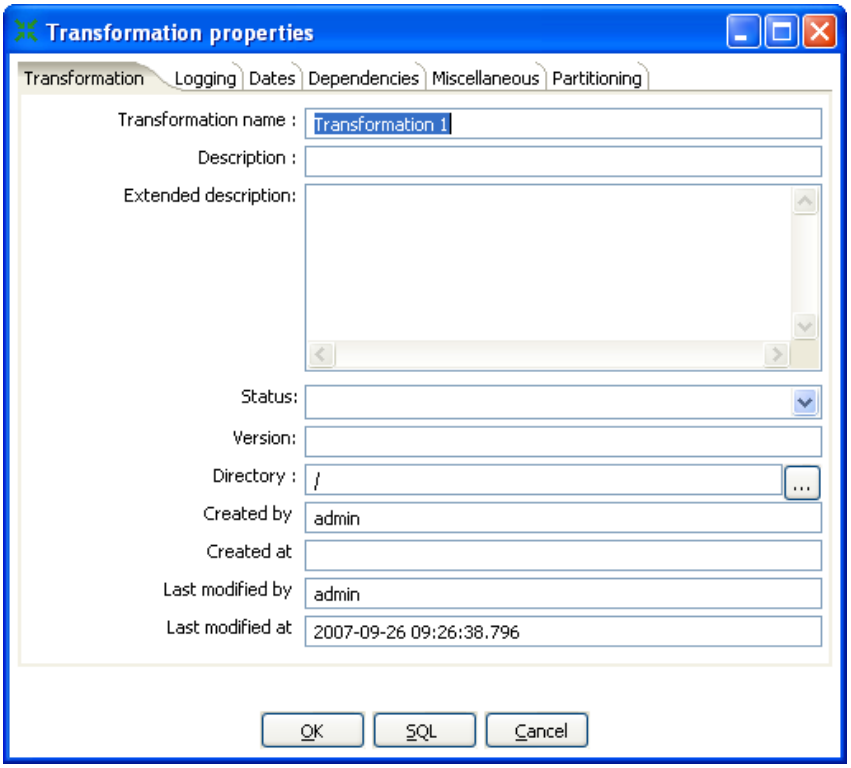
9.2.1. Environment variables

The first usage (and only usage in previous Kettle versions) was to set an environment variable. This was traditionally done by passing options to the Java Virtual Machine (JVM) with the -D option. It's also an easy way to specify the location of temporary files in a platform independent way, for example using variable ${java.io.tmpdir}. This variable points to directory /tmp on Unix/Linux/OSX and to C:\Documents and Settings\<username\Local Settings\Temp on Windows machines.

The only problem with using environment variables is that the usage is not dynamic in nature and problems arise if you would try to use them in a dynamic way. That is because if you run two or more transformations or jobs run at the same time on an application server (for example the Pentaho platform) you would get conflicts. Changes to the environment variables are visible to all software running on the virtual machine.

9.2.2. Kettle variables

Because the scope of an environment variable (9.2.1.Environment variables) is too broad, Kettle variables



Transformation Settings

were introduced to provide a way to define variables that are local to the job in which the variable is set. The "Set Variable" step in a transformation allows you to specify in which job you want to set the variable's scope (i.e. parent job, grand-parent job or the root job).

9.2.3. Internal variables

The following variables are always defined:

Variable Name	Sample value
Internal.Kettle.Build.Date	2007/05/22 18:01:39
Internal.Kettle.Build.Version	2045
Internal.Kettle.Version	2.5.0

These variables are defined in a transformation:

Variable Name	Sample value
Internal.Transformation.Filename.Directory	D:\Kettle\samples
Internal.Transformation.Filename.Name	Denormaliser - 2 series of key-value pairs.ktr
Internal.Transformation.Name	Denormaliser - 2 series of key-value pairs sample
Internal.Transformation.Repository.Directory	/

These are the internal variables that are defined in a Job:

Variable Name	Sample value
Internal.Job.Filename.Directory	/home/matt/jobs
Internal.Job.Filename.Name	Nested jobs.kjb
Internal.Job.Name	Nested job test case
Internal.Job.Repository.Directory	/

These variables are defined in a transformation running on a slave server, executed in clustered mode:

Variable Name	Sample value
Internal.Slave.Transformation.Number	0..<cluster size-1> (0,1,2,3 or 4)
Internal.Cluster.Size	<cluster size> (5)

10. Transformation Settings

10.1. Description

Transformation Settings are a collection of properties to describe the transformation and configure its behavior. Access Transformation Settings from the main menu under Transformation|Settings. The following sections provides a detailed description of the available settings.

10.2. Transformation Tab

The transformation tab allows you to specify general properties about the transformation including:

Setting	Description
Transformation name	The name of the transformation Required information if you want to save to a repository
Description	Short description of the transformation, shown in the repository explorer
Extended description	Long extended description of the transformation
Status	Draft or production status
Version	Version description
Directory	The directory in the repository where the transformation is stored
Created by	Displays the original creator of the transformation.
Created at	Displays the date and time when the transformation was created.
Last modified by	Displays the user name of the last user that modified the transformation.
Last modified at	Displays the date and time when the transformation was last modified.

10.3. Logging

The Logging tab allows you to configure how and where logging information is captured. Settings include:

Setting	Description
READ log step	Use the number of read lines from this step to write to the log table. Read means: read from source steps.
INPUT log step	Use the number of input lines from this step to write to the log table. Input means: input from file or database.
WRITE log step	Use the number of written lines from this step to write to the log table. Written means: written to target steps.
OUTPUT log step	Use the number of output lines from this step to write to the log table. Output means: output to file or database.
UPDATE log step	Use the number of updated lines from this step to write to the log table. Update means: updated in a database.
REJECTED log step	Use the number of rejected lines from this step to write to the log table. Rejected means: error record.
Log connection	The connection used to write to a log table.
Log table	specifies the name of the log table (for example L_ETL)
Use Batch-ID?	Enable this if you want to have a batch ID in the L_ETL file. Disable for backward compatibility with Spoon/Pan version < 2.0.

Setting	Description
Use logfield to store logging in	This option stores the logging text in a CLOB field in the logging table. This allows you to have the logging text together with the run results in the same table. Disable for backward compatibility with Spoon/Pan version < 2.1

10.4. Dates

The Dates tab allows you to configure the following date related settings:

Setting	Description
Maxdate connection	Get the upper limit for a date range on this connection.
Maxdate table	Get the upper limit for a date range in this table.
Maxdate field	Get the upper limit for a date range in this field.
Maxdate offset	Increases the upper date limit with this amount. Use this for example, if you find that the field DATE_LAST_UPD has a maximum value of 2004-05-29 23:00:00, but you know that the values for the last minute are not complete. In this case, simply set the offset to -60.
Maximum date difference	Sets the maximum date difference in the obtained date range. This will allow you to limit job sizes.

10.5. Dependencies

The Dependencies tab allows you to enter all of the dependencies for the transformation. For example, if a dimension is depending on 3 lookup tables, we have to make sure that these lookup tables have not changed. If the values in these lookup tables have changed, we need to extend the date range to force a full refresh of the dimension. The dependencies allow you to look up whether a table has changed in case you have a "data last changed" column in the table.

The 'Get dependencies button' will try to automatically detect dependencies.

10.6. Miscellaneous

The Miscellaneous tab allows you to configure the following settings:

Setting	Description
Number of rows in rowsets	This option allows you to change the size of the buffers between the connected steps in a transformation. You will rarely/never need to change this parameter. Only when you run low on memory it might be an option to lower this parameter.
Show a feedback row in transformation steps?	This controls whether or not to add a feedback entry into the log file while the transformation is being executed. By default, this feature is enabled and configured to display a feedback record every 5000 rows.
The feedback size	Sets the number of rows to process before entering a feedback entry into the log. Set this higher when processing large amounts of data to reduce the amount of information in the log file.
Use unique connections	This allows use to open one unique connection per defined and used database connection in the transformation. Checking this option is required in order to allow a failing transformation to be rolled back completely.
Shared objects file	Specifies the location of the XML file used to stored shared objects like database connections, clustering schemas and more.

Manage thread priorities?	Allows you to enable or disable the internal logic for changing the Java thread priorities based on the number of input and output rows in the perspective "rowset" buffers. This can be useful in some simplistic situations where the cost of using the logic is exceeds the benefit of the thread prioritization.

10.7. Partitioning

The Partitioning tab provides a list of available database partitions. You can create a new partition by clicking on the "New" button. The "Get Partitions" button will retrieve a list of available partitions that have been defined for the connection.

10.8. SQL Button

Click the SQL button at the bottom of the Transformation properties button to generate the SQL needed to create the logging table. The DDL will display in the Simple SQL Editor allowing you to execute this or any other SQL statement(s) against the logging connection.

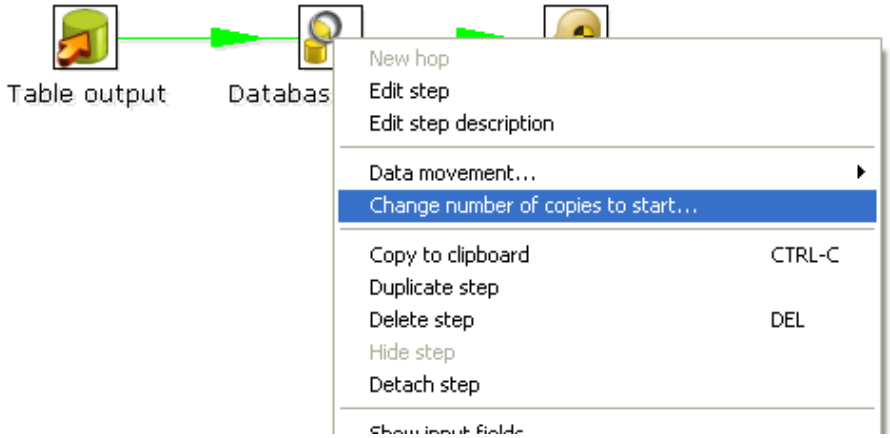
11. Transformation Steps

11.1. Description

A step is one part of a transformation. Steps can provide you with a wide range of functionality ranging from reading text-files to implementing slowly changing dimensions. This chapter describes various step settings followed by a detailed description of available step types.

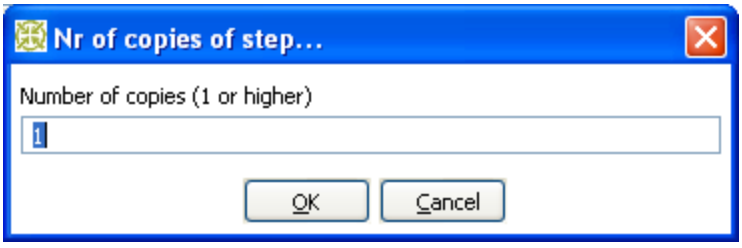
11.2. Launching several copies of a step

Sometimes it can be useful to launch the same step several times. For example, for performance reasons it can be useful to launch a database lookup step 3 times or more. That is because database connections usually have a certain latency. Launching the same step several times keeps the database busy on different connections, effectively lowering the latency. You can launch several copies of step in a transformation simply by right-clicking on a step in the graphical view and then by selecting "change number of copies to start…":



The "Step copies" popup menu

You will get this dialog:

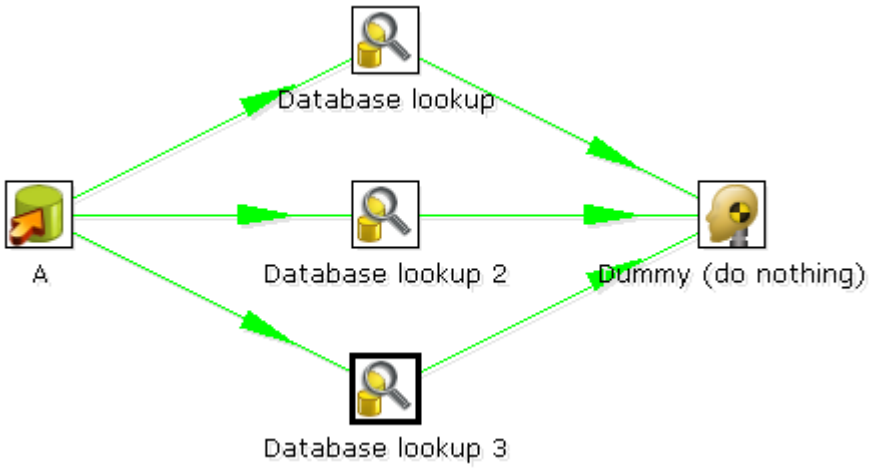


The step copies dialog

If you enter 3 this will be shown:

Multiple step copies example

It is the technical equivalent of this:

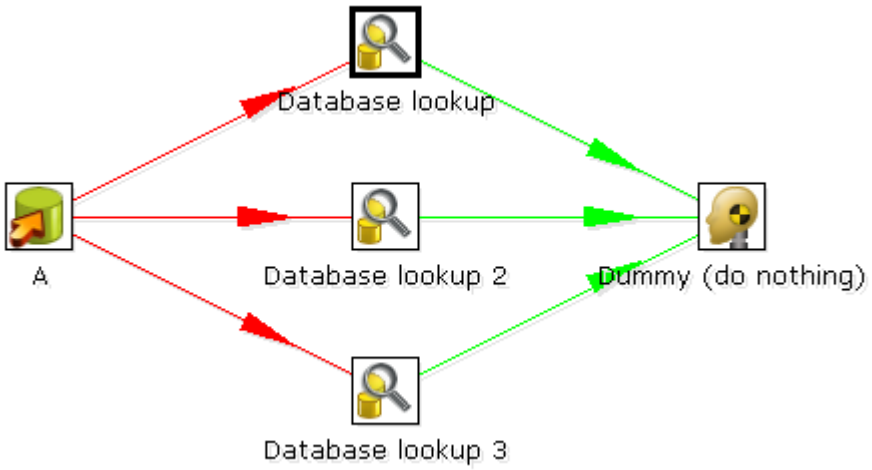


Multiple step copies equivalent

11.3. Distribute or copy?

In the example above, green lines are shown between the steps. This indicates that rows are distributed among the target steps. In this case, it means that the first row coming from step "A" goes to step "database lookup 1", the second to "database lookup 2", the third to "Database lookup 3", the fourth back to "database lookup 1", etc.

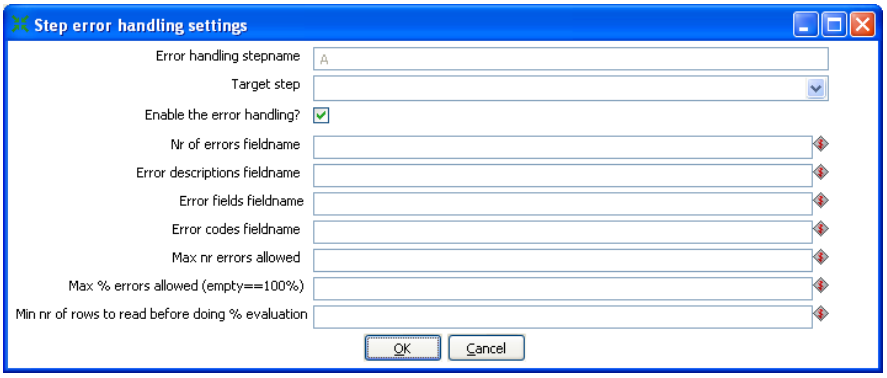
However, if we right click on step "A", and select "Copy data", you will get the hops drawn in red:



"Copy data" means that all rows from step "A" are copied to all 3 the target steps.
In this case it means that step "B" gets 3 copies of all the rows that "A" has sent out.

NOTE: *Because of the fact that all these steps are run as different threads, the order in which the single rows arrive at step "B" is probably not going to be the same as they left step "A".*

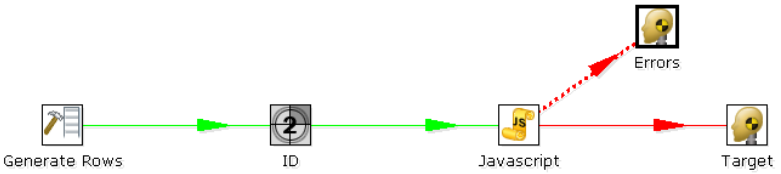
11.4. Step error handling

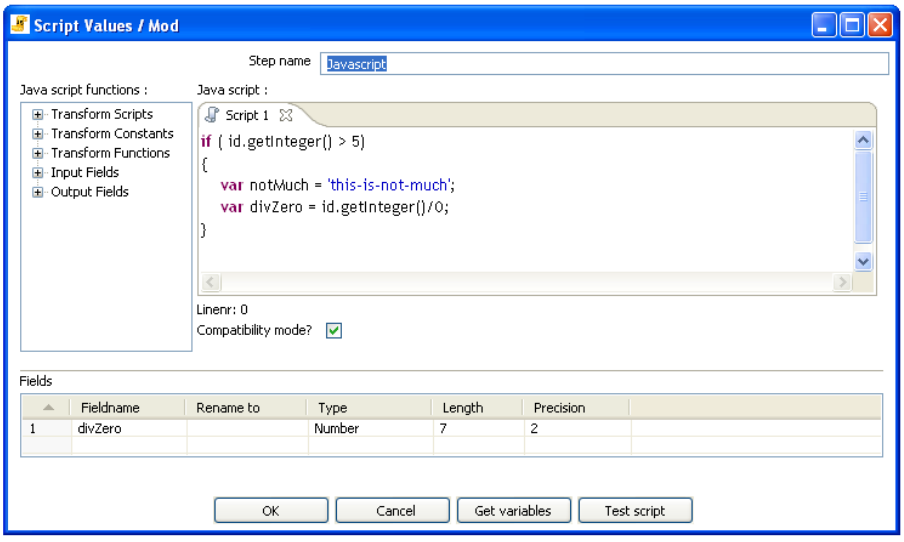


Step error handling settings

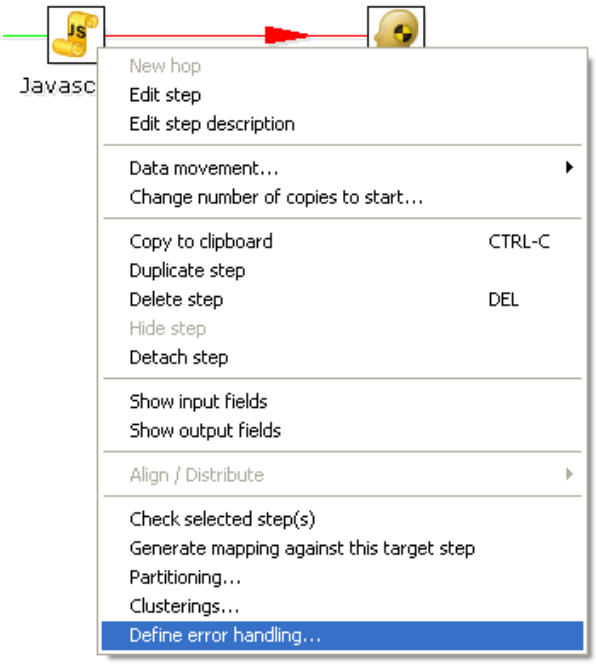
Step error handling allows you to configure a step such that instead of halting a transformation when an error occurs, pass those rows that caused an error to a different step. To configure error handling, right click on the step and select "Define Error handling...".

In the example below, we artificially generate an error in the Script Values step when an ID is higher than 5.



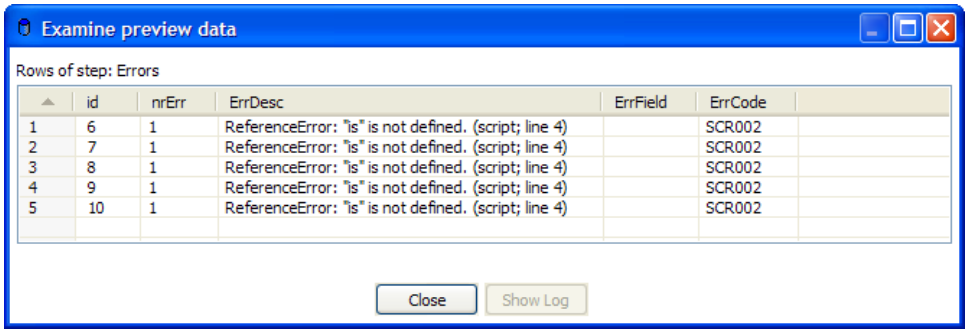


To configure the error handling, you can right click on the step involved and select the "Error handing..." menu item:



NOTE: *this menu item only appears when clicking on steps that support the new error handling code.*

As you can see, you can add extra fields being to the "error rows":



This way, we can easily define new data flows in our transformations. The typical use-case for this is an alternative way of doing an Upsert (Insert/Update):

This transformation performs an insert regardless of the content of the table. If you put a primary key on the ID (in this case the customer ID) the insert into the table cause an error. Because of the error handling we can pass the rows in error to the update step. Preliminary tests have shown this strategy of doing upserts to be 3 times faster in certain situations. (with a low updates to inserts ratio)

11.5. Apache Virtual File System (VFS) support

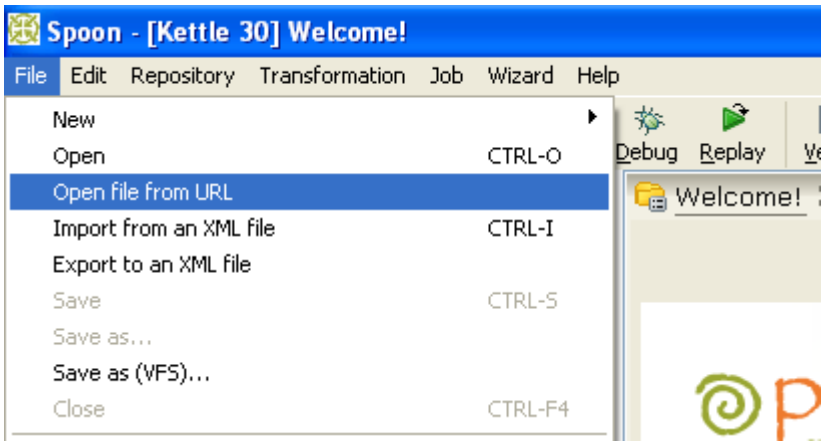
Kettle provides support for the Apache Virtual File System (VFS) as an additional way to reference source files, transformations and jobs from any location you like. For more information about VFS, visit Apache Commons Virtual File System.

11.5.1. Example: Referencing remote job files

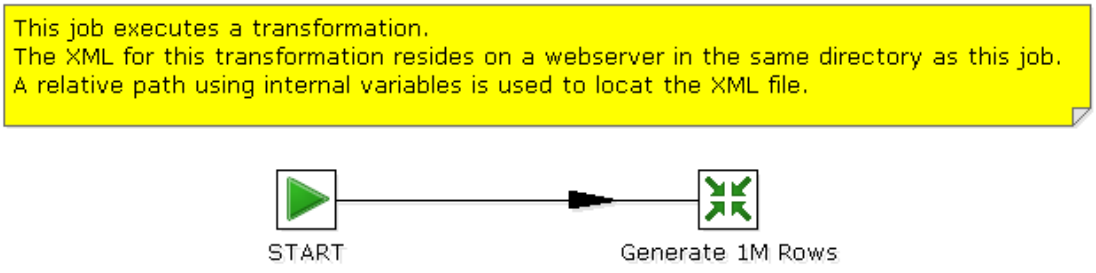
Here is a simple example of using VFS to reference the location of a job file we want to execute using Kitchen:

```
sh kitchen.sh -file:http://www.kettle.be/GenerateRows.kjb
```

To open this job using VFS from within Spoon, select File|Open file from URL:



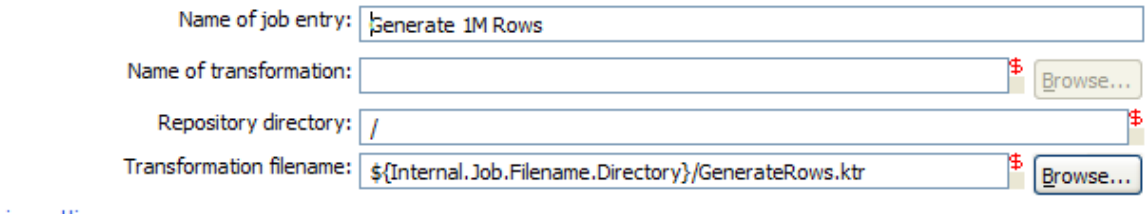
Enter the URL 'http://www.kettle.be/GenerateRows.kjb' and click the OK button to load the job in Spoon:



The transformation we are about to launch is also located on the web server. The internal variable for the job name directory is:

```
Internal.Job.Filename.Directory    http://www.kettle.be/
```

This allows us to reference the transformation as follows:

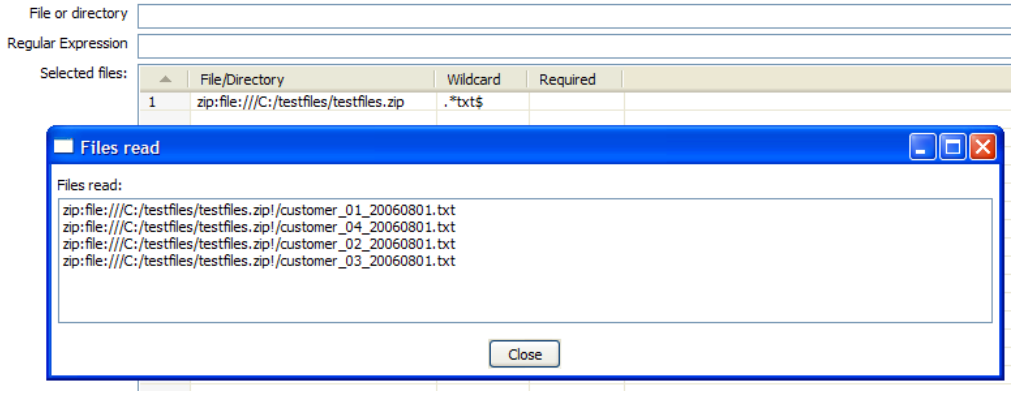


Note: *You will not be able to save the job back to the web server in this example. That is not because we do not support it, but because you don't have the permission to do so.*

For more information on the almost endless list of possibilities with VFS, please visit: http://jakarta.apache.org/commons/vfs/filesystems.html. Examples include direct loading from zip-files, gz-files, jar-files, ram drives, SMB, (s)ftp, (s)http, etc.

We will extend this list even further in the near future with our own drivers for the Pentaho solutions repository and later on for the Kettle repository (something like: psr:// and pdi:// URIs).

11.5.2. Example: Referencing files inside a Zip



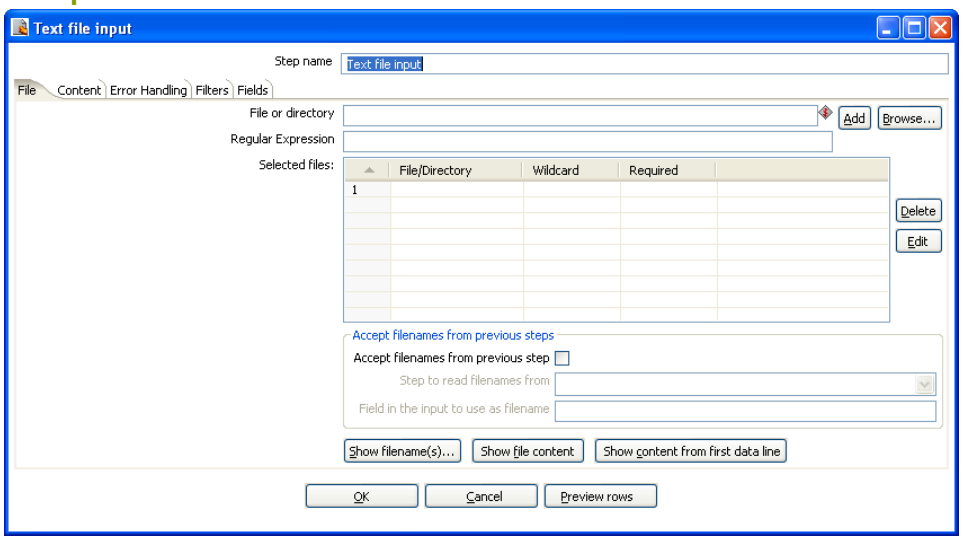
The example above illustrates the ability to use a wild-card to directly select files inside of a zip file.

Apache VFS support was implemented in all steps and job entries that are part of the Pentaho Data Integration suite as well as in the recent Pentaho platform code and in Pentaho Analyses (Mondrian).

11.6. Transformation Step Types

11.6.1. Text File Input

Icon



Text file input Dialog

11.6.1.1. General description

The Text File Input step is used to read date from a variety of different text-file types. The most commonly used formats include Comma Separated Values (CSV files) generated by spreadsheets and fixed width flat files.

The Text File Input step provides the ability to specify a list of files to read, or a list of directories with wild cards in the form of regular expressions. In addition, you can accept filenames from a previous step making filename handling more even more generic.

The following sections describe in detail the available options for configuring the Text file input step.

11.6.1.2. File options

The table below provides a detailed descriptions of the features available on the File tab:

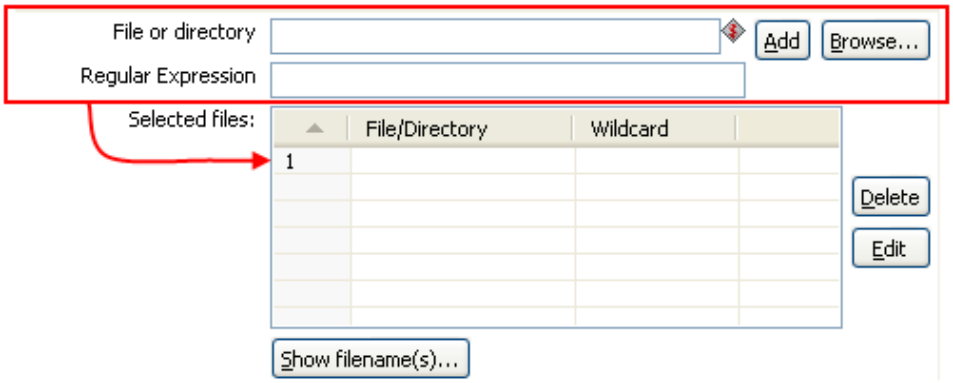
Option	Description
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Show filenames(s)...	Displays a list of all files that will be loaded based on the current selected file definitions.

Option	Description
Show file content	Displays the content of the selected file.
Show content from first data line	Displays the content from the first data line only for the selected file.

11.6.1.2.1. Selecting Files to read data from

The file tab (shown above) is where you identify the file or files from which you want to read data. To specify a file:

1. Enter the location of the file in the 'File or directory' field or click the Browse button to browse the local file system.
2. Click the 'Add' button to add a file to the list of 'selected files' like this:



Adding entries to the list of files

11.6.1.2.2. Selecting file using Regular Expressions

You can also have this step search for files by specifying a wild card in the form of a regular expression. Regular expressions are more sophisticated than simply using '*' and '?' wild cards.

Here are a few examples of regular expressions:

Filename	Regular	Expression Files selected
/dirA/	.*userdata.*\.txt	All files in /dirA/ with names containing userdata and ending on .txt
/dirB/	AAA.*	All files in /dirB/ with names starting out with AAA
/dirC/	[A-Z][0-9].*	All files in /dirC/ with names starting with a capital and followed by a digit (A0-Z9)

11.6.1.2.3. Accept filenames from previous step

Accepting filenames from previous steps

This option allows even more flexibility in combination with other steps like "Get Filenames". You can construct your filename and pass it to this step. This way the filename can come from any source: text file, database table, etc.

Option	Description
Accept filenames from previous steps	This enables the option to get filenames from previous steps.
Step to read filenames from	The step to read the filenames from
Field in the input to use as filename	Text File Input will look in this step to determine the filenames to use.

11.6.1.3. Content specification

The content tab allows you to specify the format of the text files that are being read. Here is a list of the options on this tab:

Option	Description
File type	This can be either CSV or Fixed length. Based on this selection, Spoon will launch a different helper GUI when you press the "get fields" button in the last "fields" tab.
Separator	One or more characters that separate the fields in a single line of text. Typically this is ; or a tab.
Enclosure	Some fields can be enclosed by a pair of strings to allow separator characters in fields. The enclosure string is optional. If you use repeat an enclosures allow text line `'Not the nine o''clock news.'`. With ' the enclosure string, this gets parsed as `Not the nine o'clock news`.
Allow breaks in enclosed fields?	This is an experimental feature which is currently disabled. **Note:** *This functionality is implemented and available in the CSV Input Step.*
Escape	Specify an escape character (or characters) if you have escaped characters in your data. If you have \ as an escape character, the text `'Not the nine o\'clock news.'` (with ' the enclosure) will get parsed as `Not the nine o'clock news`.
Header & number of header lines	Enable this option if your text file has a header row. (First lines in the file) You can specify the number of times the header lines appears.
Footer & number of footer lines	Enable this option if your text file has a footer row. (Last lines in the file) You can specify the number of times the footer row appears.
Wrapped lines & number of wraps	Use this if you deal with ***data*** lines that have wrapped beyond a certain page limit. Note that headers & footers are never considered wrapped.
Paged layout & page size & doc header	You can use these options as a last resort when dealing with texts meant for printing on a line printer. Use the number of document header lines to skip introductory texts and the number of lines per page to position the data lines.
Compression	Enable this option if your text file is placed in a Zip or GZip archive. **NOTE**: *At the moment, only the first file in the archive is read.*
No empty rows	Don't send empty rows to the next steps.
Include filename in output	Enable this if you want the filename to be part of the output.
Filename field name	The name of the field that contains the filename.
Rownum in output?	Enable this if you want the row number to be part of the output.
Row number field name	The name of the field that contains the row number.
Rownum by file?	Allows the row number to be reset per file.

Option	Description
Format	This can be either DOS, UNIX or mixed. UNIX files have lines that are terminated by line feeds. DOS files have lines separated by carriage returns and line feeds. If you specify mixed, no verification is done.
Encoding	Specify the text file encoding to use. Leave blank to use the default encoding on your system. To use Unicode specify UTF-8 or UTF-16. On first use, Spoon will search your system for available encodings.
Limit	Sets the number of lines that is read from the file. 0 means: read all lines.
Be lenient when parsing dates?	Disable this option if you want strict parsing of data fields. In case lenient parsing is enabled, dates like Jan 32nd will become Feb 1st.
The date format Locale	This locale is used to parse dates that have been written in full like "February 2nd, 2006". Parsing this date on a system running in the French (fr_FR) locale would not work because February would be called Février in that locale.

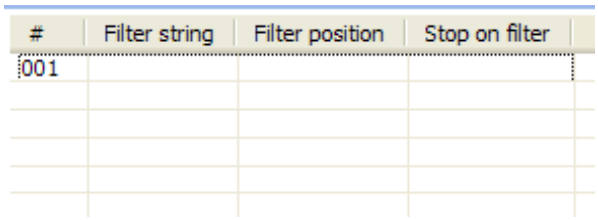
11.6.1.4. Error handling

The error handling tab was added to allow you to specify how this step should react when errors occur. The table below describes the options available for Error handling:

Option	Description
Ignore errors?	Check this option if you want to ignore errors during parsing
Skip error lines	Enable this option if you want to skip those lines that contain errors. Note that you can generate an extra file that will contain the line numbers on which the errors occurred. If lines with errors are not skipped, the fields that did have parsing errors, will be empty (null)
Error count field name	Add a field to the output stream rows. This field will contain the number of errors on the line.
Error fields field name	Add a field to the output stream rows. This field will contain the field names on which an error occurred.
Error text field name	Add a field to the output stream rows. This field will contain the descriptions of the parsing errors that have occurred.
Warnings file directory	When warnings are generated, they will be put in this directory. The name of that file will be <warning dir>/filename.<date_time>.<warning extension>
Error files directory	When errors occur, they will be put in this directory. The name of that file will be <errorfile_dir>/filename.<date_time>.<errorfile_extension>
Failing line numbers files directory	When a parsing error occur on a line, the line number will be put in this directory. The name of that file will be <errorline dir>/filename.<date_time>.<errorline extension>

11.6.1.5. Filters

The filters tab provides the ability to specify the lines you want to skip in the text file.



Specifying text file filters

The table below describes the available options for defining filters:

Option	Description
Filter string	The string to look for.
Filter position	The position where the filter string has to be at in the line. 0 is the first position in the line. If you specify a value below 0 here, the filter string is searched for in the entire string.
Stop on filter	Specify Y here if you want to stop processing the current text file when the filter string is encountered.

11.6.1.6. Fields

The fields tab is where you specify the information about the name and format of the fields being read from the text file. Available options include:

Option	Description
Name	name of the field
Type	Type of the field can be either String, Date or Number
Format	See Number Formats for a complete description of format symbols.
Length	For *Number*: Total number of significant figures in a number; For *String:* total length of string; For *Date:* length of printed output of the string (e.g. 4 only gives back the year).
Precision	For *Number:* Number of floating point digits; For *String, Date, Boolean:* unused;
Currency	used to interpret numbers like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10;000.00) or "," (5.000,00)
Grouping	A grouping can be a dot "," (10;000.00) or "." (5.000,00)
Null if	treat this value as NULL
Default	The default value in case the field in the text file was not specified. (empty)
Trim	type trim this field (left, right, both) before processing
Repeat	Y/N: If the corresponding value in this row is empty: repeat the one from the last time it was not empty

11.6.1.6.1. Number Formats

The information on Number formats was taken from the Sun Java API documentation, to be found here: http://java.sun.com/j2se/1.4.2/docs/api/java/text/DecimalFormat.html

Symbol	Location	Localized	Meaning
0	Number	Yes	Digit
#	Number	Yes	Digit, zero shows as absent
.	Number	Yes	Decimal separator or monetary decimal separator
-	Number	Yes	Minus sign
,	Number	Yes	Grouping separator
E	Number	Yes	Separates mantissa and exponent in scientific notation. Need not be quoted in prefix or suffix.
;	Sub pattern boundary	Yes	Separates positive and negative sub patterns
%	Prefix or suffix	Yes	Multiply by 100 and show as percentage
\u2030	Prefix or suffix	Yes	Multiply by 1000 and show as per mille
¤ (\u00A4)	Prefix or suffix	No	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or suffix	No	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".

Scientific Notation

In a pattern, the exponent character immediately followed by one or more digit characters indicates scientific notation. Example: "0.###E0" formats the number 1234 as "1.234E3".

11.6.1.6.2. Date formats

The information on Date formats was taken from the Sun Java API documentation, to be found here: http://java.sun.com/j2se/1.4.2/docs/api/java/text/SimpleDateFormat.html

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number 0	
k	Hour in day (1-24)	Number 24	
K	Hour in am/pm (0-11)	Number 0	
h	Hour in am/pm (1-12)	Number 12	
m	Minute in hour	Number 30	
s	Second in minute	Number 55	
S	Millisecond	Number 978	
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

11.6.1.7. Extras

Function/Button	Description
Show filenames	This option shows a list of all the files selected. Please note that if the transformation is to be run on a separate server, the result might be incorrect.
Show file content	The "View" button shows the first lines of the text-file. Make sure that the file-format is correct. When in doubt, try both DOS and UNIX formats.
Show content from first data line	This button helps you in positioning the data lines in complex text files with multiple header lines, etc.
Get fields	This button allows you to guess the layout of the file. In case of a CSV file, this is done pretty much automatically. When you selected a file with fixed length fields, you need to specify the field boundaries using a wizard.
Preview rows	Press this button to preview the rows generated by this step.

11.6.2. Table input

Icon

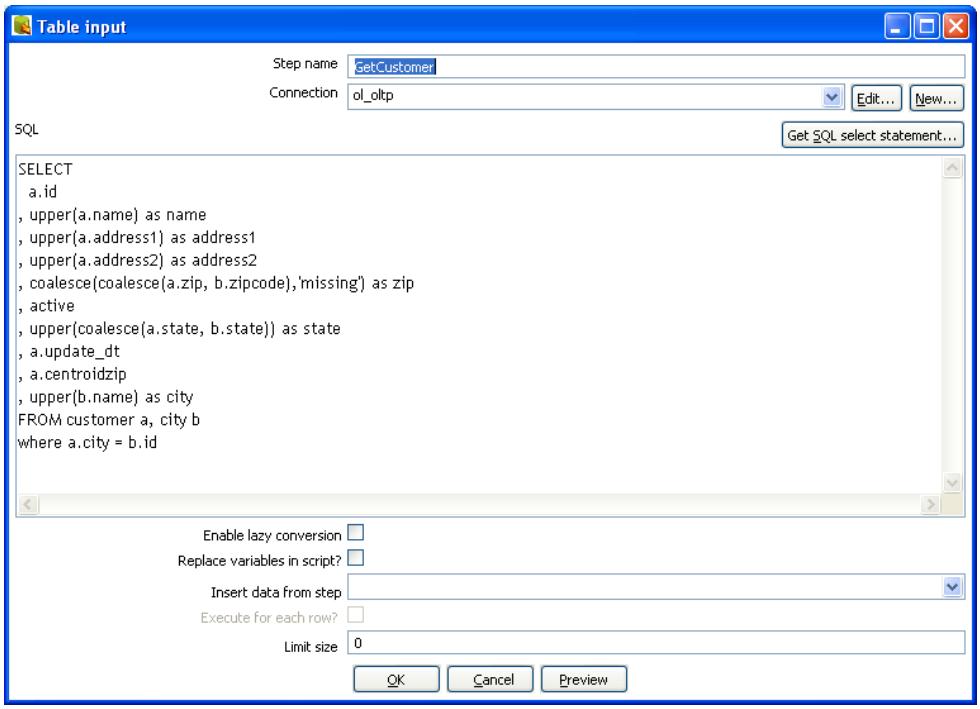


Table Input

11.6.2.1. General description

This step is used to read information from a database, using a connection and SQL. Basic SQL statements are generated automatically.

11.6.2.2. Options

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Connection	The database connection used to read data from.
SQL	The SQL statement used to read information from the database connection. You can also click the 'Get SQL select statement...' button to browse tables and automatically generate a basic select statement.
Enable lazy conversion	Lazy conversion will avoid unnecessary data type conversions and can result in a significant performance improvements. Check to enable.
Replace variables in script?	Enable this to replace variables in the script. This feature was provided to allow you to test with or without performing variable substitutions.
Insert data from step	Specify the input step name where we can expect information to come from. This information can then be inserted into the SQL statement. The locators where we insert information is indicated by ? (question marks).
Execute for each row?	Enable this option to perform the data insert for each individual row.
Limit size	Sets the number of lines that is read from the database. 0 means: read all lines.

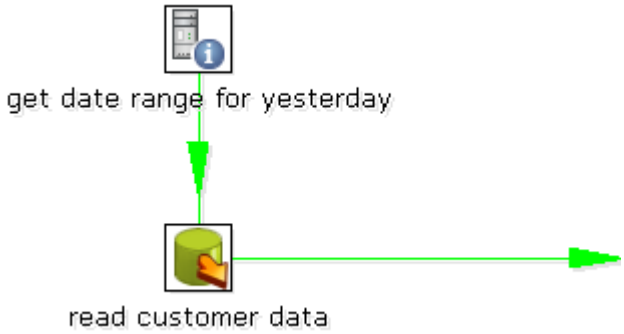
11.6.2.3. Example:

Consider for example the following SQL statement:

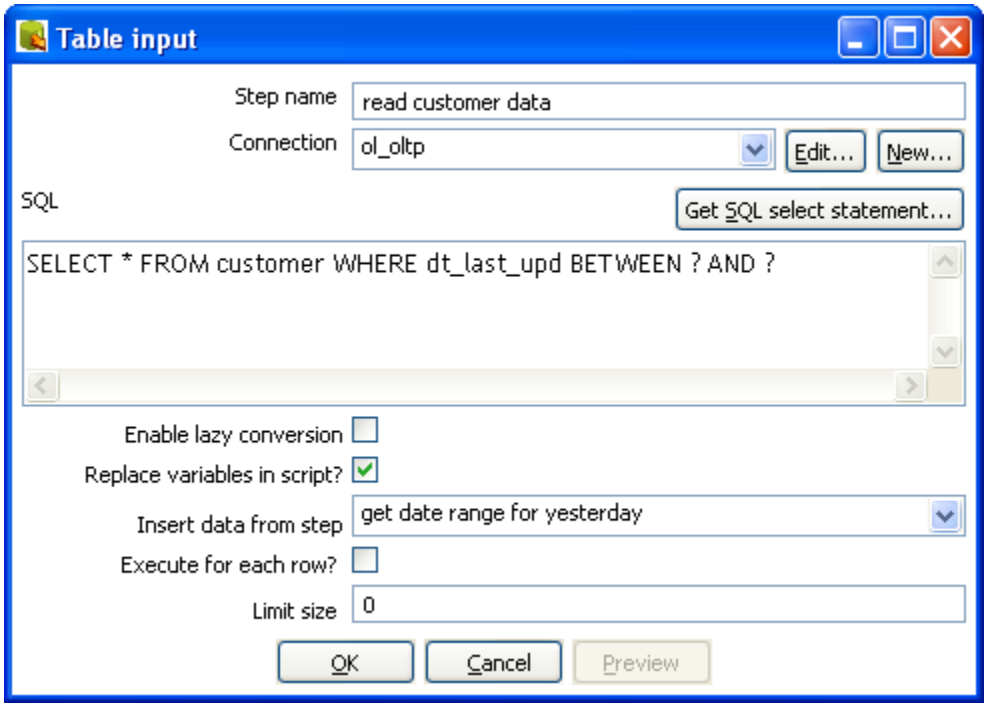
```
SELECT * FROM customers WHERE changed_date BETWEEN ? AND ?
```

This statement needs 2 dates that are read on the "Insert data from" step.

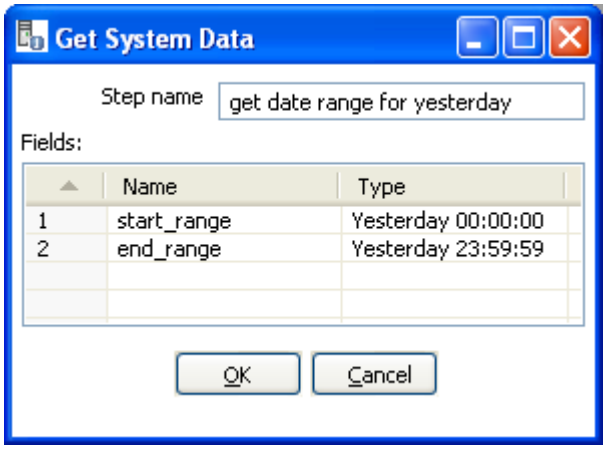
NOTE: *The dates can be provided using the "Get System Info" step type. For example if you want to read all customers that have had their data changed yesterday, you might do it like this:*



The "read customer data" step looks like this:



And the "get date range for yesterday" looks like this:

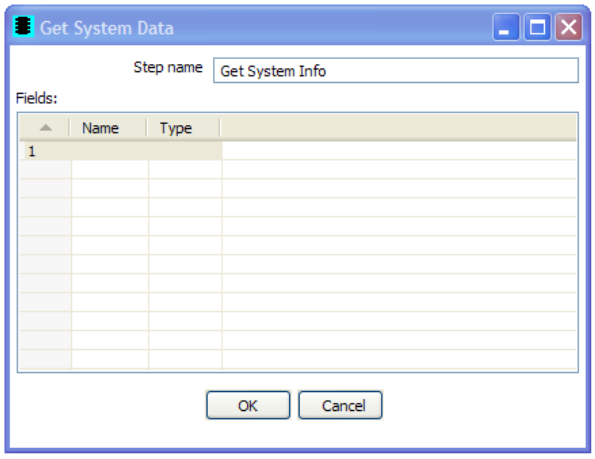


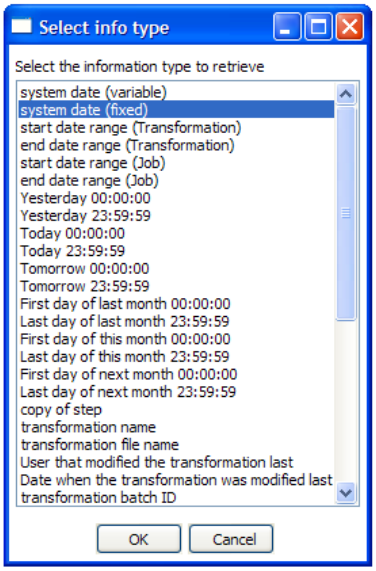
11.6.2.4. Extras

Function/Button	Description
Preview	This option previews this step. It is done by preview of a new transformation with 2 steps: this one and a Dummy step. You can see the detailed logging of that execution by clicking on the logs button in the the preview window.

11.6.3. Get System Info

Icon





11.6.3.1. General description

This step retrieves information from the Kettle environment, available information to retrieve includes:

Item	Description
system date (variable)	System time, changes every time you ask a date.
system date (fixed)	System time, determined at the start of the transformation.
start date range (Transformation)	Start of date range, based upon information in ETL log table. See, also Transformation Settings.
end date range (Transformation)	End of date range, based upon information in ETL log table. See, also Transformation Settings.
start data range (Job)	Start of date range based upon information in the ETL log table. See also Transformation Settings.
End date range (Job)	End of date range based upon information in the ETL log table. See also Transformation Settings.
Yesterday 00:00:00	Start of yesterday.
Yesterday 23:59:59	End of yesterday.
Today 00:00:00	Start of today.
Today 23:59:59	End of today.
Tomorrow 00:00:00	Start of tomorrow.
Tomorrow 23:59:59	End of tomorrow
First day of last month 00:00:00	Start of last month.
Last day of last month 23:59:59	End of last month.
First day of this month 00:00:00	Start of this month.
Last day of this month 23:59:59	End of this month.
First day of next month 00:00:00	Start of next month.
Last day of next month 23:59:59	End of next month.
copy of step	Copy nr of the step. See also Launching several copies of a

Item	Description
	step.
transformation name	Name of the transformation.
transformation file name	File name of the transformation (XML only).
User that modified the transformation last	
Date when the transformation was modified last	
transformation batch ID	ID_BATCH value in the logging table, see 6. Transformation settings.
Hostname	Returns the hostname of the server.
IP address Returns the IP address of the server.	
command line argument 1	Argument 1 on the command line.
command line argument 2	Argument 2 on the command line.
command line argument 3	Argument 3 on the command line.
command line argument 4	Argument 4 on the command line.
command line argument 5	Argument 5 on the command line.
command line argument 6	Argument 6 on the command line.
command line argument 7	Argument 7 on the command line.
command line argument 8	Argument 8 on the command line.
command line argument 9	Argument 9 on the command line.
command line argument 10	Argument 10 on the command line.
Kettle version	Returns the Kettle version (e.g. 2.5.0)
Kettle Build Version	Returns the build version of the core Kettle library (e.g. 13)
Kettle Build Date	Returns the build date of the core Kettle library

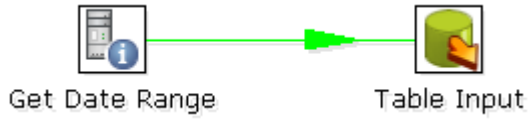
11.6.3.2. Options

There following table describes the options for configuring the Get System info step:

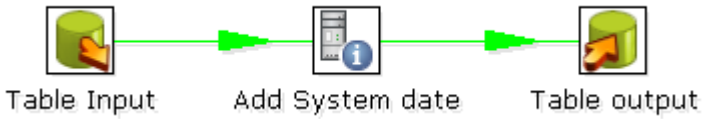
Option	Description
Step Name	Name of the step. This name has to be unique in a single transformation.
Fields	The fields to output.

11.6.3.3. Usage

The first type of usage is to simply get information from the system:

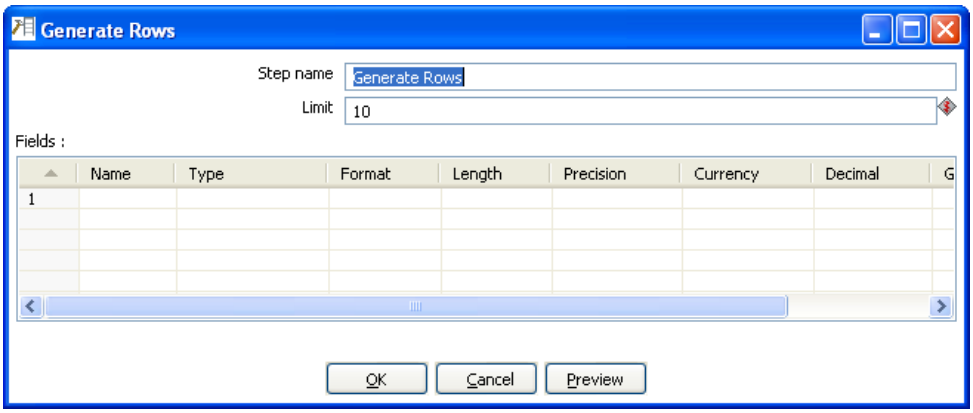


From version 2.3.0 on, this step also accepts input rows. The selected values will be added to the rows found in the input stream(s):



11.6.4. Generate Rows

Icon



Generate Rows

11.6.4.1. General description

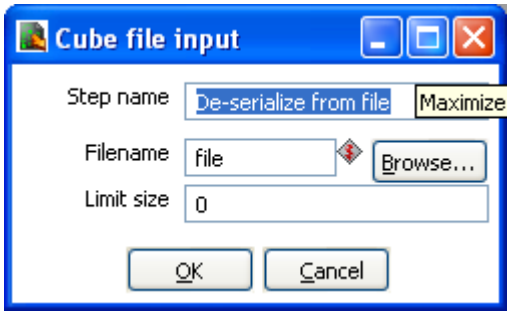
This step type outputs a number of rows, default empty but optionally containing a number of static fields.

11.6.4.2. Options

Option	Description
Step Name	Name of the step. This name has to be unique in a single transformation.
Limit	Sets the maximum number of rows you want to generate.
Fields	This table is where you configure the structure and values (optionally) of the rows you are generating.

11.6.5. De-serialize from file (formerly Cube Input)

Icon



De-serialze from file

11.6.5.1. General description

Read rows of data from a binary Kettle cube file.

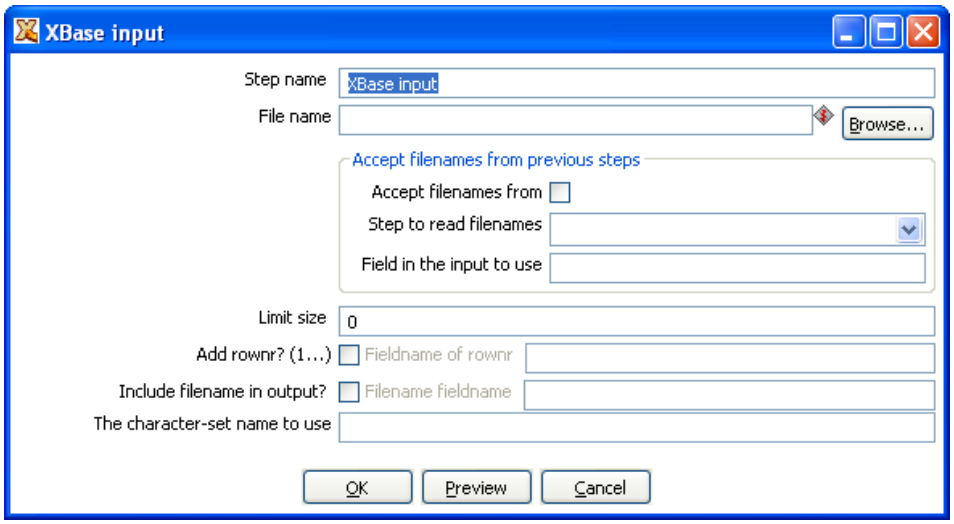
NOTE: *This step should only be used to store short lived data. It is not guaranteed that the file format stays the same between versions of Pentaho Data Integration.*

11.6.5.2. Options

Option	Description
Step Name	Name of the step. This name has to be unique in a single transformation.
Filename	The name of the Kettle cube file that will be generated.
Limit Size	Allows you to optionally limit the number of rows written to the cube file. A value of '0' indicates no size limit.

11.6.6. XBase input

Icon



Xbase input

11.6.6.1. General description

With this step it is possible to read data from most types of DBF file derivates called the XBase family. (dBase III/IV, Foxpro, Clipper, ...)

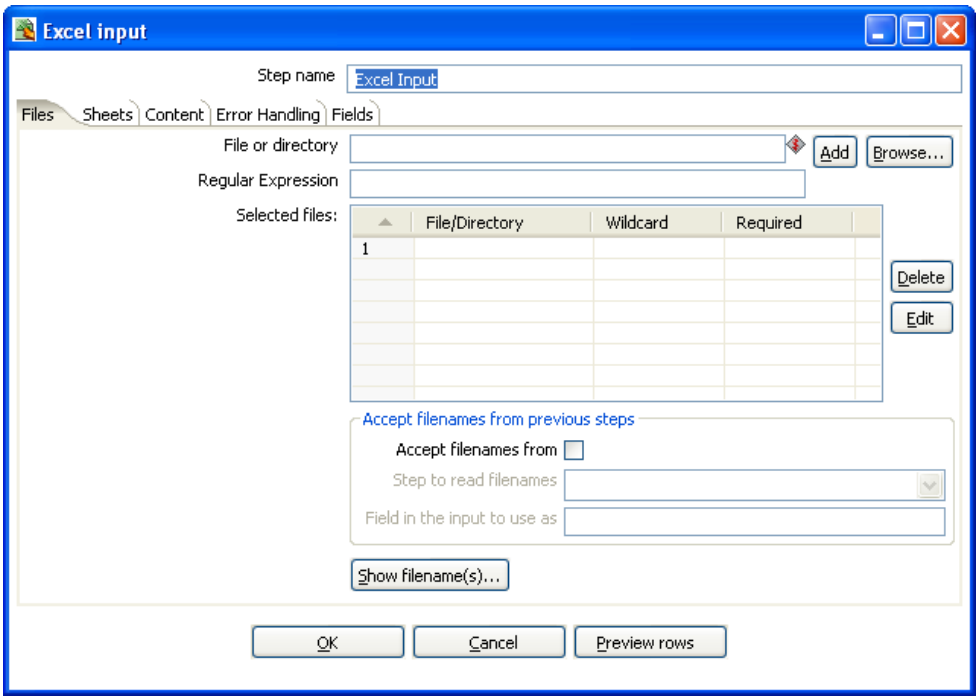
11.6.6.2. Options

The following options are available for the Xbase input step:

Option	Description
Step Name	Name of the step. This name has to be unique in a single transformation.
Filename	The name of the DBF file to read data from
Limit Size	Allows you to optionally limit the number of rows read.
Accept filenames	Allows you to read in filenames from a previous step in the transformation.
Add rownr?	Adds a field to the output with the specified name that contains the row number.
Include filename in output?	Optionally allows you to insert a field containing the filename onto the stream.
Character-set name to use	Specifies the character set (i.e. ASCII, UTF-8) to use.
Preview	Click this button to preview that will be read.

11.6.7. Excel input

Icon



Excel input

11.6.7.1. General description

This step provides the ability to read data from one or more Excel files. The following sections describe each of the available features for configuring the Excel input step.

11.6.7.2. Files Tab

The files tab is where you define the location of the Excel files you wish to read from. Available options include:

Option	Description
Step Name	Name of the step. This name has to be unique in a single transformation.
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Accept filenames from previous steps	Allows you to read in filenames from a previous step in the transformation.
Show filenames(s)...	Displays a list of all files that will be loaded based on the current

Option	Description
	selected file definitions.
Preview rows	Click to preview the contents of the specified Excel file.

11.6.7.3. Sheets

The List of sheets to read table displays currently selected sheets to read from. Use the 'Get sheetname(s)' button to fill in the available sheets automatically.

Note: *You also need to specify the start row and column for each selected sheet. This determines the coordinates for where the step should start reading.*

11.6.7.4. Content

The content tab allows you to configure the following properties:

Option	Description
Header	Check if the sheets specified have a header row that we need to skip.
No empty rows	Check this if you don't want empty rows in the output of this step.
Stop on empty row	This will make the step stop reading the current sheet of a file when a empty line is encountered.
Filename field	Specify a field name to include the filename in the output of this step.
Sheetname field	Specify a field name to include the sheetname in the output of this step.
Sheer row nr field	Specify a field name to include the sheet row number in the output of the step. The sheet row number is the actual row number in the Excel sheet.
Row nrwritten field	Specify a field name to include the row number in the output of the step. "Row number written" is the number of rows processed, starting at 1 and counting up regardless of sheets and files.
Limit	limit the number of rows to this number, 0 means: all rows.
Encoding	Specify the character encoding (i.e. UTF-8, ASCII)

11.6.7.5. Error handling

The Error handling tab allows you to configure the following properties:

Option	Description
Strict types?	Enable this option if you want to fail immediately upon reading an unexpected field type. When disabled, Kettle will attempt to convert incoming fields to the requested data type.
Ignore errors?	Check this option if you want to ignore errors during parsing
Skip error lines?	Enable this option if you want to skip those lines that contain errors. **Note:** *you can generate an extra file that will contain the line numbers on which the errors occurred. If lines with errors are not skipped, the fields that did have parsing errors, will be empty (null).*
Warnings file directory	When warnings are generated, they will be put in this directory. The name of that file will be <warning

Option	Description
	dir>/filename.<date_time>.<warning extension>
Error files directory	When errors occur, they will be put in this directory. The name of that file will be <errorfile_dir>/filename.<date_time>.<errorfile_extension>
Failing line numbers files directory	When a parsing error occur on a line, the line number will be put in this directory. The name of that file will be <errorline dir>/filename.<date_time>.<errorline extension>

11.6.7.6. Fields

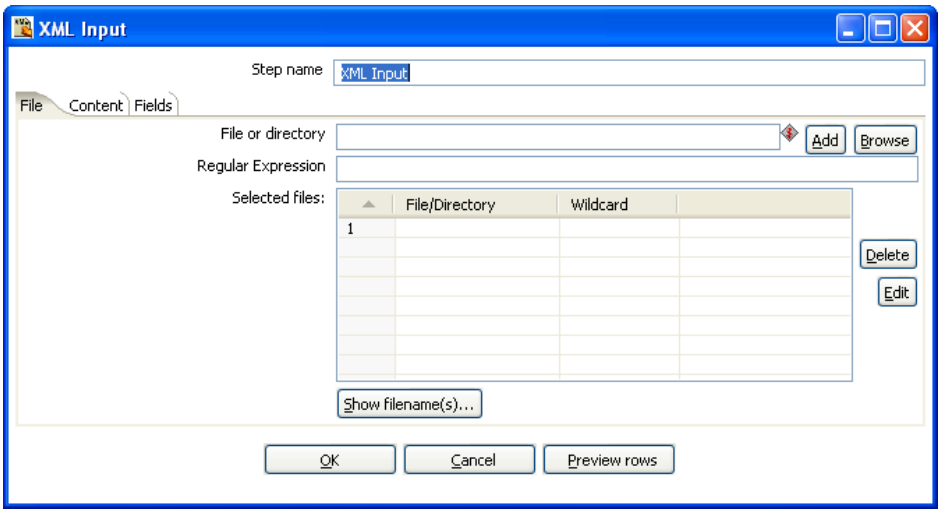
The fields tab is for specifying the fields that need to be read from the Excel files. A button "Get fields from header row" is provided to automatically fill in the available fields if the sheets have a header row.

For a given field, the 'Type' column is provided for performing data type conversions. For example, if you want to read a Date and you have a String value in the Excel file, you can specify the conversion mask.

Note: *in the case of Number to Date conversion (example: 20051028 --> October 28th, 2005) you should simply specify the conversion mask yyyyMMdd because there will be an implicit Number to String conversion taking place before doing the String to Date conversion.*

11.6.8. XML Input

Icon



XML Input dialog

11.6.8.1. General description

This step allows you to read information stored in XML files. The following sections descibe the interface for defining the filenames you want to read from, the repeating part of the data part of the XML file and the fields to retrieve.

Note: *You specify the fields by the path to the Element or Attribute and by entering conversion masks, data types and other meta-data.*

11.6.8.2. File Tab

The files tab is where you define the location of the Excel files you wish to read from. Available options include:

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Show Filename(s)	This option shows a list of the files the will be generated. **NOTE**: *This is a simulation and sometimes depends on the number of rows in each file, etc.*

11.6.8.3. Content

The content tab contains the following options for describing the content being read:

Option	Description
Include filename in output & fieldname	Check this option if you want to have the name of the XML file to which the row belongs in the output stream. You can specify the name of the field where the filename will end up in.
Rownum in output & fieldname	Check this option if you want to have a row number (starts at 1) in the output stream. You can specify the name where the integer will end up in.
Limit	You can specify the maximum number of rows to read here.
Nr of header rows to skip	Specify the number of rows to skip, from the start of an XML document, before starting to process.
Location	Specify the path by way of elements to the repeating part of the XML file. For example if you are reading rows from this XML file: `<Rows>` `  <Row>` `  <Field1>...</Field1> ...` `  </Row>` `  ...` `</Rows>` Then you set the location to Rows, Row **Note:** *you can also set the root (Rows) as a repeating element location. The output will then contain 1 (one) row.*

11.6.8.4. Fields

The fields tab is where you define properties for the location and format of the fields being read from the XML document. The table below describes each of the options for configuring the field properties:

Option	Description
Name	The name of the field.
Type	Type of the field can be either String, Date or Number.
Format	The format mask to convert with. See Number Formats for a complete description of format specifiers.
Length	The length option depends on the field type follows: • Number - Total number of significant figures in a number • String - total length of string • Date - length of printed output of the string (e.g. 4 only gives back year)
Precision	The precision option depends on the field type as follows: • Number - Number of floating point digits • String - unused • Date - unused
Currency	Symbol used to represent currencies like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10,000.00) or "," (5.000,00)
Group	A grouping can be a "," (10,000.00) or "." (5.000,00)

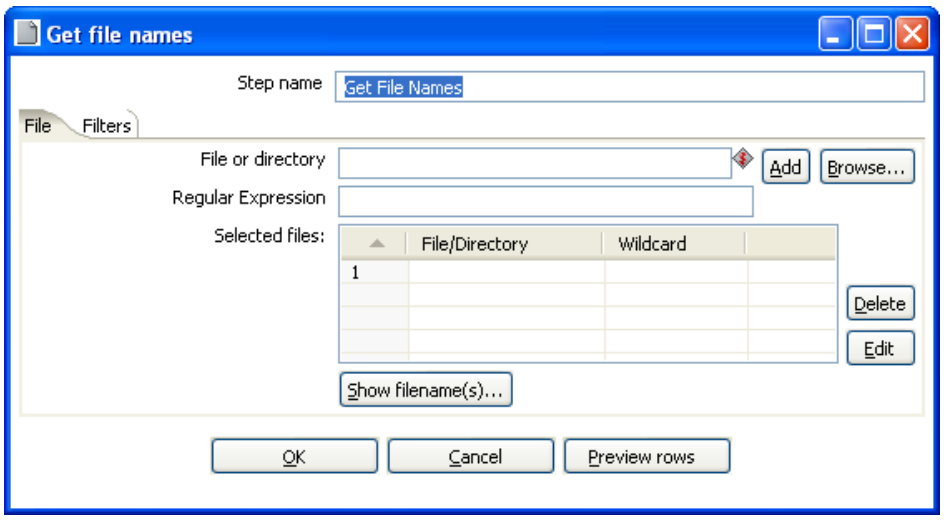
Option	Description
Trim type	The trimming method to apply on the string found in the XML.
Repeat	Check this if you want to repeat empty values with the corresponding value from the previous row.
Position	The position of the XML element or attribute. You use the following syntax to specify the position of an element, for example: The first element called "element": `E=element/1` The first attribute called "attribute": `A=attribute/1` The first attribute called "attribute" in the second "element" tag: `E=element/2, A=attribute/1`

NOTE: *You can auto-generate all the possible positions in the XML file supplied by using the "Get Fields" button.*

NOTE: *Support was added for XML documents where all the information is stored in the Repeating (or Root) element. The special R= locater was added to allow you to grab this information. The "Get fields" button finds this information if it's present.*

11.6.9. Get File Names

Icon



Get File Names dialog

11.6.9.1. General description

This step allows you to get information regarding filenames on the file system. The retrieved filenames are added as rows onto the stream.

The output fields for this step are:

- filename - the complete filename, including the path (/tmp/kettle/somefile.txt)
- short_filename - only the filename, without the path (somefile.txt)
- path - only the path (/tmp/kettle/)

11.6.9.1.1. File tab

This tab defines the location of the files you want to retrieve filenames for. For more information about specifying file locations, see Selecting Files to read data from.

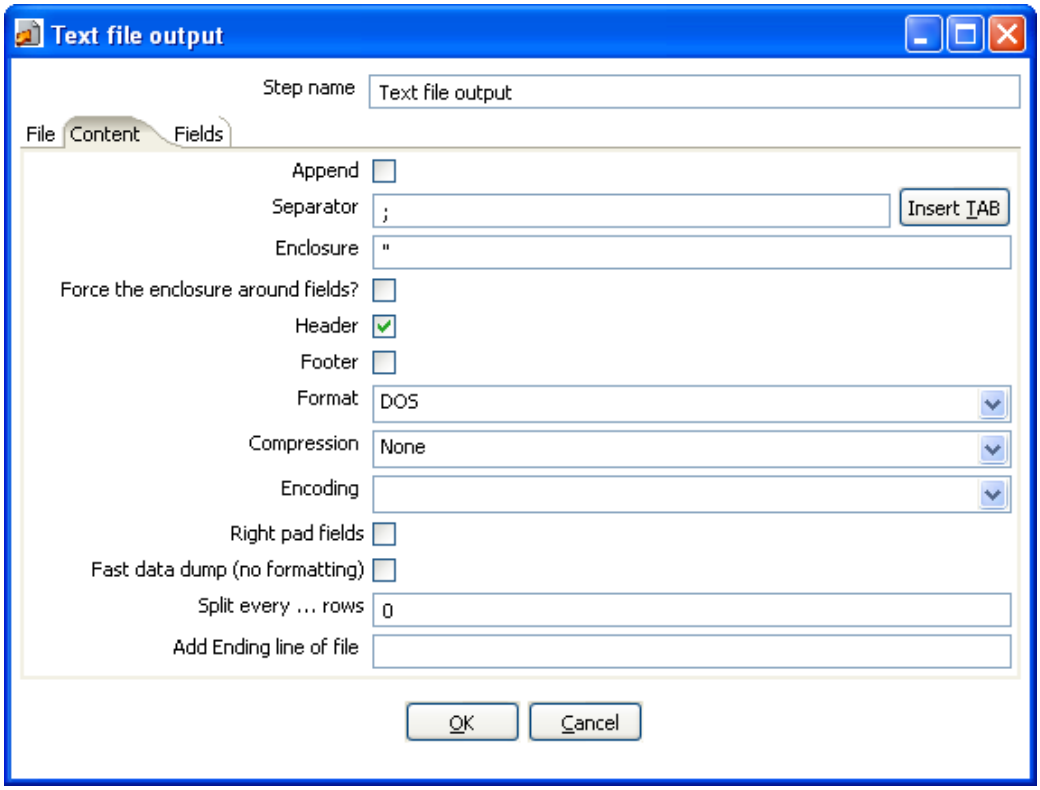
11.6.9.1.2. Filters

The filters tab allows you to filter the retrived filenames based on:

- All files and folders
- Files only
- Folders only

11.6.10. Text File Output

Icon



Text file output dialog

11.6.10.1. General Description

The Text file output step is used to export data to text file format. Tis is commonly used to generate Comma Separated Values (CSV files) that can be read by spreadsheet applications.

11.6.10.2. File Tab

The File tab is where you define basic properties about the file being created, such as:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Filename	This field specifies the filename and location of the output text file.
Run this as a command instead?	Check this to "pipe" the results into the command or script you specify.
Extension	Adds a point and the extension to the end of the filename. (.txt)
Include stepnr in filename	If you run the step in multiple copies (Launching several copies of a step), the copy number is included in the filename, before the extension. (_0).
Include partition nr in filename?	Includes the data partition number in the filename.
Include date in filename	Includes the system date in the filename. (_20041231).

Option	Description
Include time in filename	Includes the system date in the filename. (_235959).
Show filename(s)	This option shows a list of the files the will be generated. **NOTE**: *This is a simulation and sometimes depends on the number of rows in each file, etc.*

11.6.10.3. Content

The content tab contains the following options for describing the content being read:

Option	Description
Append	Check this to append lines to the end of the specified file.
Separator	Specify the character that separates the fields in a single line of text. Typically this is ; or a tab.
Enclosure	A pair of strings can enclose some fields. This allows separator characters in fields. The enclosure string is optional. Header Enable this option if you want the text file to have a header row. (First line in the file).
Force the enclosure around fields?	This option forces all field names to be enclosed with the character specified in the Enclosure property above.
Header	Enable this option if you want the text file to have a header row. (First line in the file).
Footer	Enable this option if you want the text file to have a footer row. (Last line in the file).
Format	This can be either DOS or UNIX. UNIX files have lines are separated by linefeeds. DOS files have lines separated by carriage returns and line feeds.
Encoding	Specify the text file encoding to use. Leave blank to use the default encoding on your system. To use Unicode specify UTF-8 or UTF-16. On first use, Spoon will search your system for available encodings.
Compression	Allows you to specify the type of compression, .zip or .gzip to use when compressing the output. **NOTE**: *At the moment, only one file is placed in a single archive.*
Right pad fields	Add spaces to the end of the fields (or remove characters at the end) until they have the specified length.
Fast data dump (no formatting)	Improves the performance when dumping large amounts of data to a text file by not including any formatting information.
Split every … rows	If this number N is larger than zero, split the resulting text-file into multiple parts of N rows.
Add Ending line of file	Allows you to specify an alternate ending row to the output file.

11.6.10.4. Fields

The fields tab is where you define properties for the fields being exported. The table below describes each of the options for configuring the field properties:

Option	Description
Name	The name of the field.

Option	Description
Type	Type of the field can be either String, Date or Number.
Format	The format mask to convert with. See Number Formats for a complete description of format symbols.
Length	The length option depends on the field type follows: • Number - Total number of significant figures in a number • String - total length of string • Date - length of printed output of the string (e.g. 4 only gives back year)
Precision	The precision option depends on the field type as follows: • Number - Number of floating point digits • String - unused • Date - unused
Currency	Symbol used to represent currencies like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10,000.00) or "," (5.000,00)
Group	A grouping can be a "," (10,000.00) or "." (5.000,00)
Trim type	The trimming method to apply on the string found in the XML.
Null	If the value of the field is null, insert this string into the textfile
Get fields	Click to retrieve the list of fields from the input stream(s)
Minimal width	Alter the options in the fields tab in such a way that the resulting width of lines in the text file is minimal. So instead of save 0000001, we write 1, etc. String fields will no longer be padded to their specified length.

11.6.11. Table output

Icon

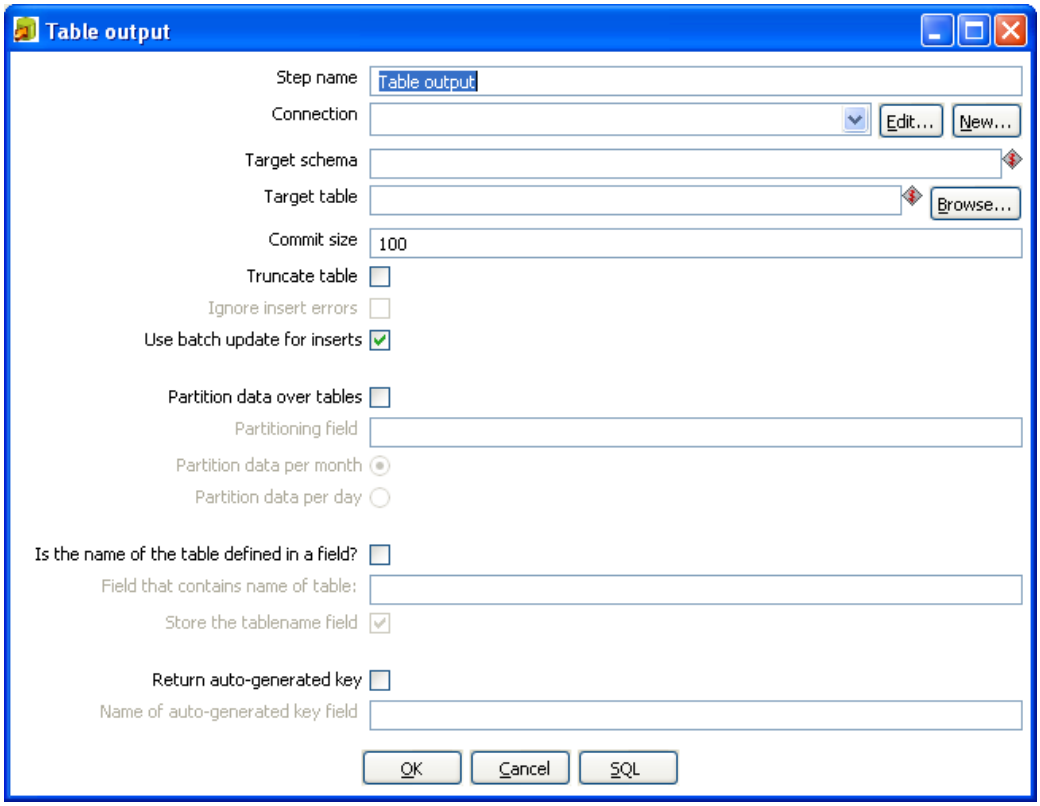


Table output dialog

11.6.11.1. General description

This step type allows you to load data into a database table.

11.6.11.2. Options

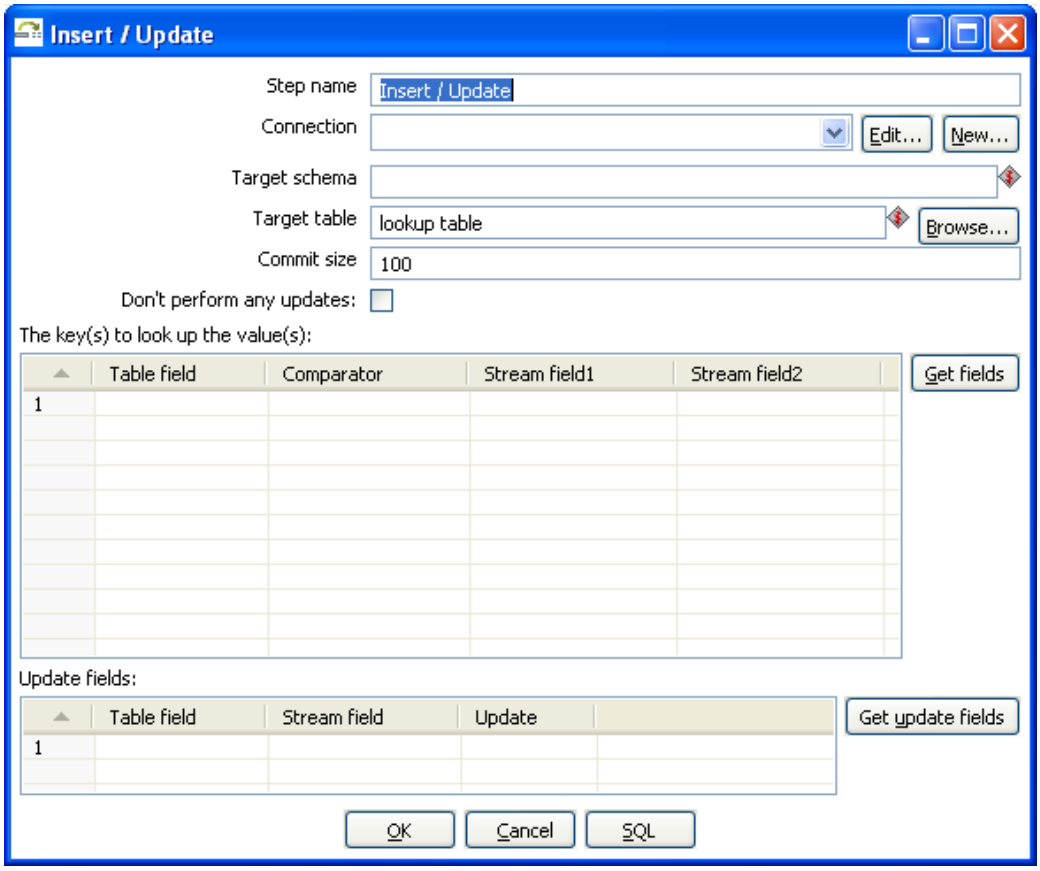
The table below describes the available options for the Table output step:

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Connection	The database connection used to write data to.
Target Schema	The name of the Schema for the table to write data to. This is important for data sources that allow for table names with dots '.' In it.
Target table	The name of the table to write data to.
Commit size	Use transactions to insert rows in the database table. Commit the connection every N rows if N is larger than 0. Otherwise, don't use transactions. (Slower) **NOTE:** *Transactions are not supported on all database platforms.*
Truncate table	Select this if you want the table to be truncated before the first row is inserted into the table.
Ignore insert errors	Makes Kettle ignore all insert errors such as violated primary keys. A maximum of 20 warnings will be logged however. This option is not

Option	Description
	available for batch inserts.
Use batch update for inserts	Enable this option if you want to use batch inserts. This feature groups inserts statements to limit round trips to the database. This is the fastest option and is enabled by default.
Partition data over tables	Use this options to split the data over multiple tables. For example instead of inserting all data into table SALES, put the data into tables SALES_200510, SALES_200511, SALES_200512, ... Use this on systems that don't have partitioned tables and/or don't allow inserts into UNION ALL views or the master of inherited tables. The view SALES allows you to report on the complete sales: CREATE OR REPLACE VIEW SALES AS SELECT * FROM SALES_200501 UNION ALL SELECT * FROM SALES_200502 UNION ALL SELECT * FROM SALES_200503 UNION ALL SELECT * FROM SALES_200504 ...
Is the name of the table defined in a field.	Use these options to split the data over one or more tables. The name of the target table is defined in the field you specify. For example if you store customer data in the field gender, the data might end up in tables M and F (Male and Female). There is an option to exclude the field containing the tablename from being inserted into the tables.
Return auto-generated key	Check this if you want to get back the key that was generated by inserting a row into the table.
Name of auto-generated key field	Specify the name of the new field in the output rows that will contain the auto-generated key.
SQL	Generate the SQL to create the output table automatically.

11.6.12. Insert / Update

Icon



Insert/Update dialog

11.6.12.1. General description

This step type first looks up a row in a table using one or more lookup keys. If the row can't be found, it inserts the row. If it can be found and the fields to update are the same, nothing is done. If they are not all the same, the row in the table is updated.

11.6.12.2. Options

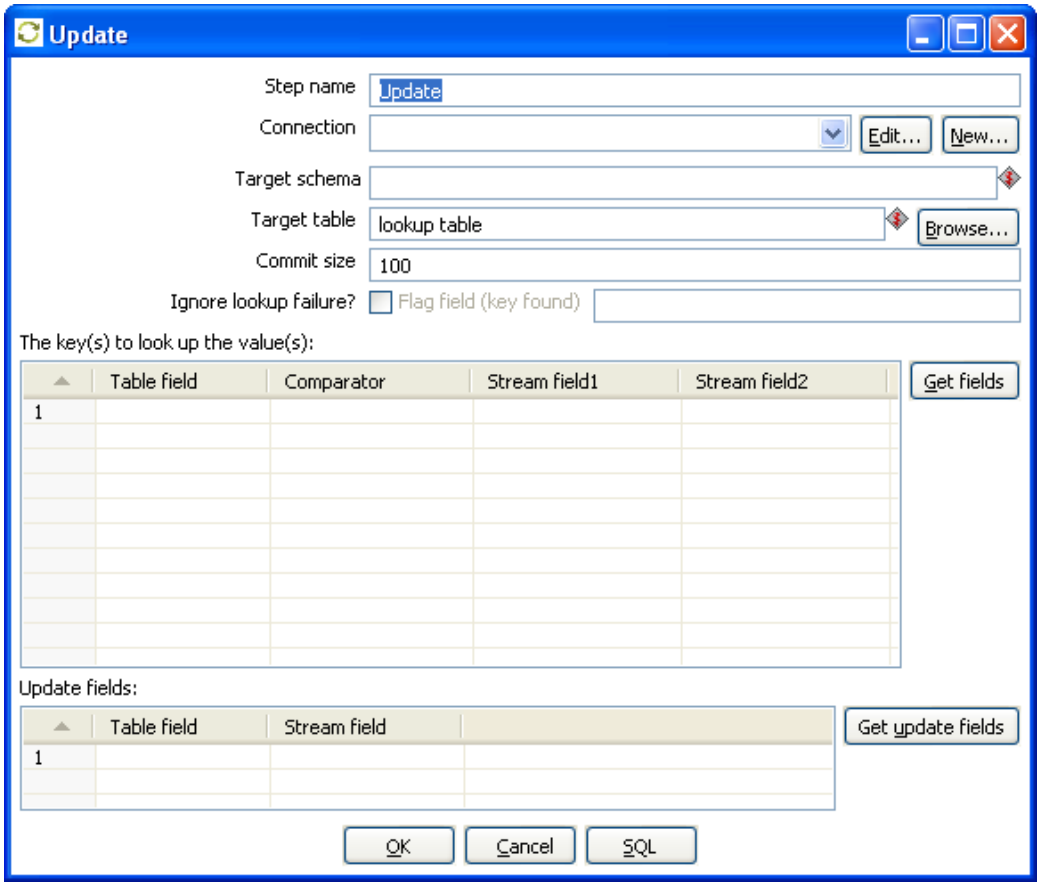
The table below provides a description of available options for Insert/Update:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Connection	The database connection used to write data to.
Target schema	The name of the Schema for the table to write data to. This is important for data sources that allow for table names with dots '.' In it.
Target table	Name of the table in which you want to do the insert or update.
Commit size	The number of rows to change (insert / update) before running a commit.
Don't perform any updates	If this option is checked, the values in the database are never updated. Only inserts are done.

Option	Description
Key Lookup table	Here you specify a list of field values and comparators. You can use the following comparators: =, <>, <, <=, >, >=, LIKE, BETWEEN, IS NULL, IS NOT NULL **Note:** *Click the 'Get fields' button to retrieve a list of fields from the input streasm(s).*
Update Fields	Specify all fields in the table you want to insert/update including the keys. Please note that you can avoid updates on certain fields by specifying N in the update column. **Note:** *Click the 'Get update fields' button to retrieve a list of update fields from the input stream(s).*
SQL button	Click the 'SQL' button to generate the SQL to create the table and indexes for correct operation.

11.6.13. Update

Icon



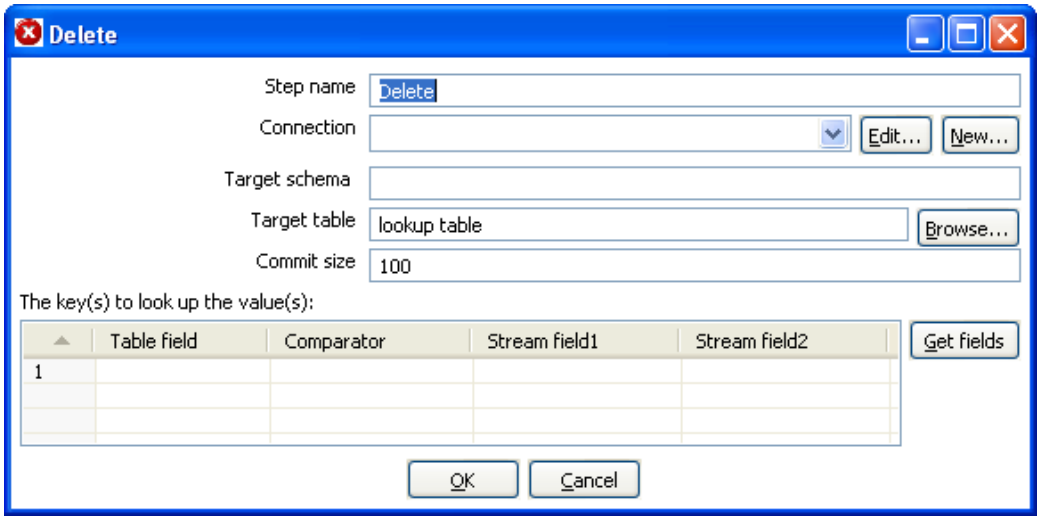
Update dialog

11.6.13.1. General description

This step is the same as the Insert / Update step except that no insert is ever done in the database table, ONLY updates are performed.

11.6.14. Delete

Icon



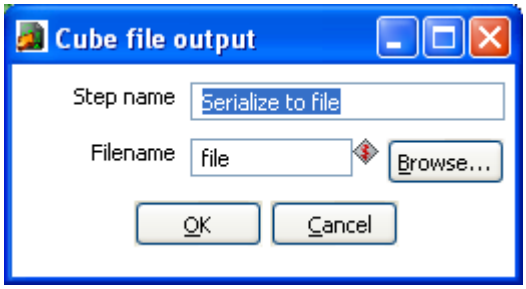
Delete dialog

11.6.14.1. General description

This step is the same as the Update step except that instead of updating, rows are deleted.

11.6.15. Serialize to file (formerly Cube File Output)

Icon



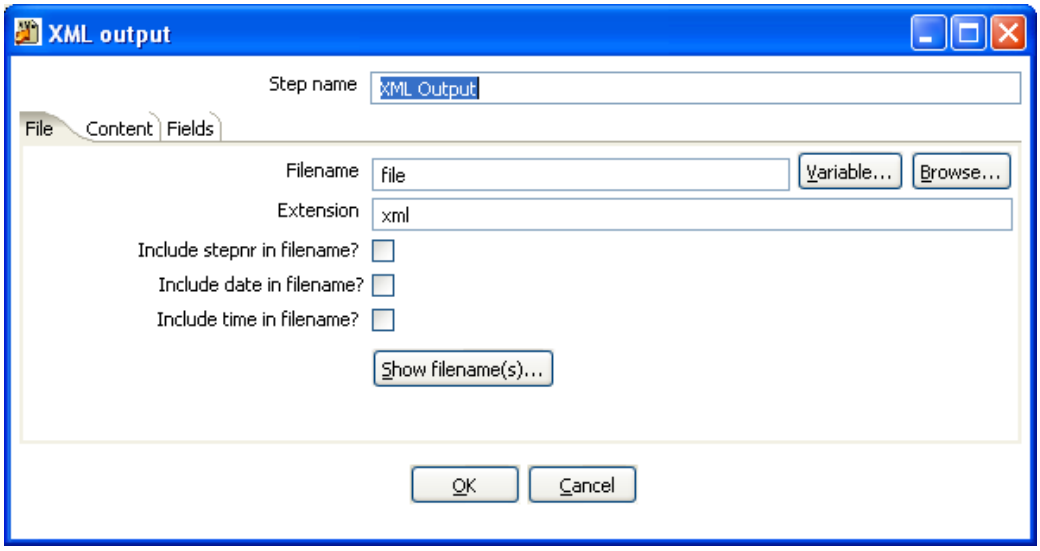
Serialize to file dialog

11.6.15.1. General description

This step stores rows of data in a binary form in a file. It has the advantage over a text (flat) file that the content does not have to be parsed when read back. This is because the meta-data is stored in the cube file as well.

11.6.16. XML Output

Icon



XML output dialog

11.6.16.1. General description

This step allows you to write rows from any source to one or more XML files.

11.6.16.2. File Tab

The file tab is where you set general properties for the XML output file format:

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Filename	This field specifies the filename and location of the output text file.
Extension	Adds a point and the extension to the end of the filename (.xml).
Include stepnr in filename	If you run the step in multiple copies (see also Launching Several Copies of a step), the copy number is included in the filename, before the extension (_0).
Include date in filename	Includes the system date in the filename (_20041231).
Include time in filename	Includes the system date in the filename (_235959).

11.6.16.3. Content

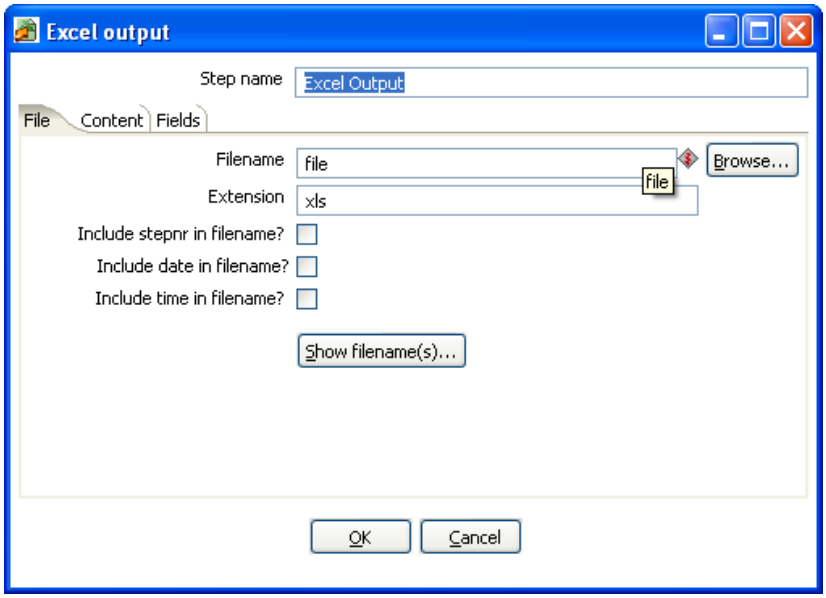
Option	Description
Zipped	Check this if you want the XML file to be stored in a ZIP archive.
Encoding	The encoding to use. This encoding is specified in the header of the XML file.
Parent XML element	The name of the root element in the XML document.
Row XML element	The name of the row element to use in the XML document.
Split every ... rows.	The maximum number of rows of data to put in a single XML file before another is created

11.6.16.4. Fields

Option	Description
Fieldname	The name of the field.
Elementname	The name of the element in the XML file to use. Type: Type of the field can be either String, Date, or Number.
Type	Type of the field can be either String, Date, or Number. Format mask to convert with: see Number formats for a complete description of format specifiers.
Length	The length option depends on the field type follows: • Number - Total number of significant figures in a number • String - total length of string • Date - length of printed output of the string (e.g. 4 only gives back year) **Note:** *the output string is padded to this length if it is specified.*
Precision	The precision option depends on the field type as follows: • Number - Number of floating point digits • String - unused • Date - unused
Currency	Symbol used to represent currencies like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10,000.00) or "," (5.000,00)
Group	A grouping can be a "," (10,000.00) or "." (5.000,00)
Null	If the value of the field is null, insert this string into the textfile
Get fields	Click to retrieve the list of fields from the input stream(s).
Minimal width	Alter the options in the fields tab in such a way that the resulting width of lines in the text file is minimal. So instead of save 0000001, we write 1, etc. String fields will no longer be padded to their specified length.

11.6.17. Excel Output

Icon



Excel output dialog

11.6.17.1. General description

With this step you can write data to one or more Excel files. The following sections describe the features available for configuring the Excel output step.

11.6.17.2. File Tab

The file tab is where you configure the filename of the Excel output step. Available options include:

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Filename	This field specifies the filename and location of the output text file.
Extension	Adds a point and the extension to the end of the filename (.xml).
Include stepnr in filename	If you run the step in multiple copies (see also Launching Several copies of a step, the copy number is included in the filename, before the extension (_0).
Include date in filename	Includes the system date in the filename (_20041231).
Include time in filename	Includes the system date in the filename (_235959).
Show filename(s)	This option shows a list of the files the will be generated. **NOTE**: *This is a simulation and sometimes depends on the number of rows in each file, etc.*

11.6.17.3. Content

The content tab provides additional options for the generated Excel output file including:

Option	Description
Header	Check if the spreadsheet needs a header above the exported rows of data.
Footer	Check if the spreadsheet needs a footer below the exported rows of data.
Encoding	Specify the encoding of the spreadsheet, leave empty to keep the default for the platform.
Split every...rows	Splits the data over several output files. (each in it's own spreadsheet)
Sheet name	Specify the name of the Sheet to write to.
Protect sheet?	Check to enable password protection on the target sheet.
Password	Specify the password for the protected sheet.
Use Template	This is an experimental feature that requires testing. Check this to use a template when outputting data to Excel.
Excel Template	The name of the template used to format the Excel output file
Append to Excel template	Check this option to have the output appended to the Excel template specified

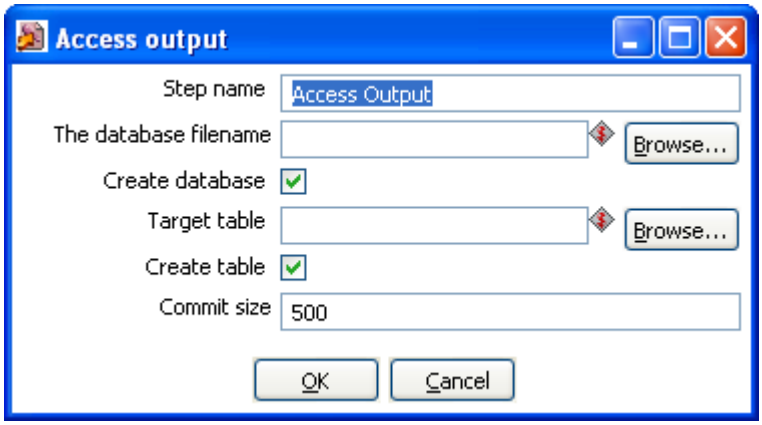
11.6.17.4. Fields

The fields tab is where you specify the Name, data type and format of the fields being written to Excel. The 'Get Fields' button will retrieve a list of available fields from the input stream(s) coming into the step. The 'Minimal Width' button will automatically alter the options in the fields tab in such a way that the resulting width of lines in the text file is minimal. So instead of save 0000001, we write 1, etc. String fields will no longer be padded to their specified length.

Note: *You can specify any format definitions available in Excel. These formats are not tied to any Kettle specific formatting.*

11.6.18. Microsoft Access Output

Icon



Microsoft Access output dialog

11.6.18.1. General Description

This allows you to create a new Access database file as an output in a transformation.

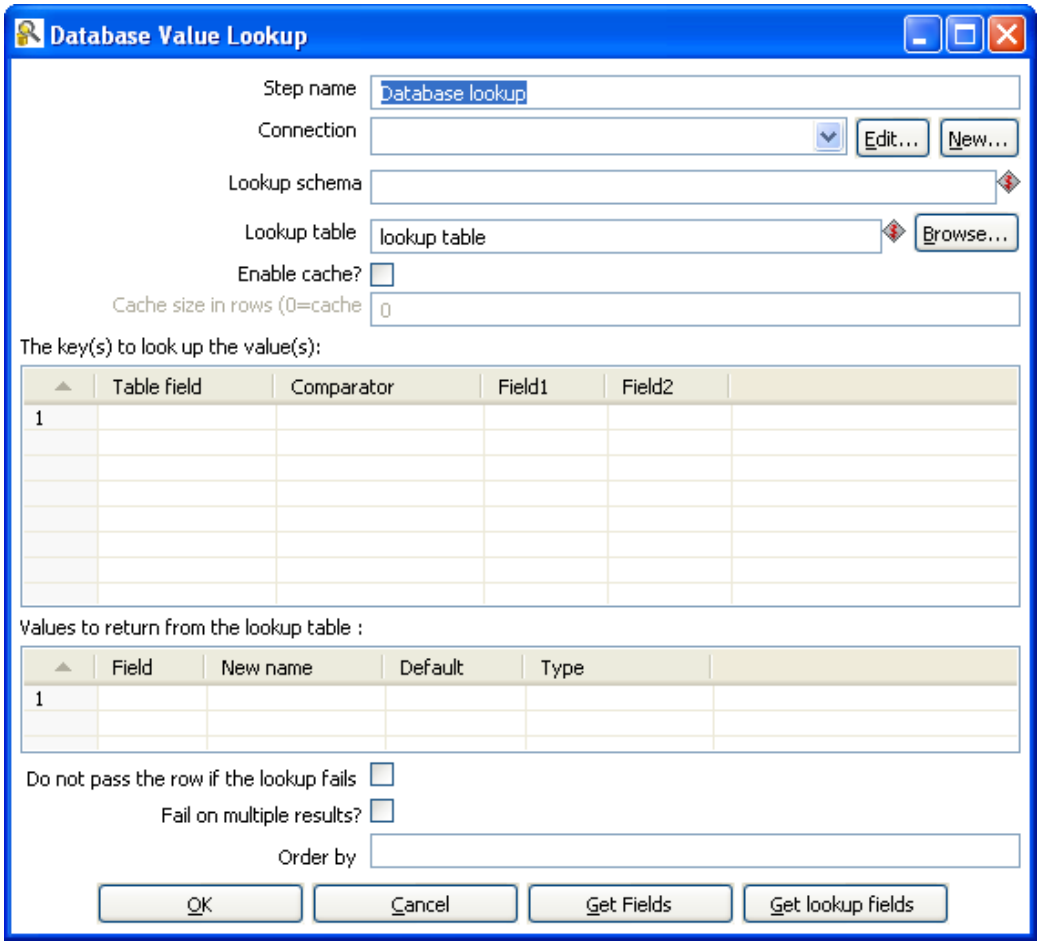
11.6.18.2. Options

The following options are available for configuring the Access output:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
The database filename	The filename of the database file you are connecting to.
Create database	Check this to generate a new Access database file.
Target table	Specify the table you want to output data to.
Create table	Check this to create a new table in the Access database.
Commit size	Defines the commit size when outputting data.

11.6.19. Database lookup

Icon



Database Value Lookup dialog

11.6.19.1. General description

This step type allows you to look up values in a database table. Lookup values are added as new fields onto the stream.

11.6.19.2. Options

The following table describes the available options for configuring the database lookup:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Connection	Select the database connection for the lookup.
Lookup schema	Specify the target schema to use for the lookup.
Lookup Table	The name of the table where we do the lookup.
Enable cache?	This option caches database lookups. This means that we expect the database to return the same value all the time for a certain lookup value.
Cache size in rows	Specify the size in rows of the cache to use.

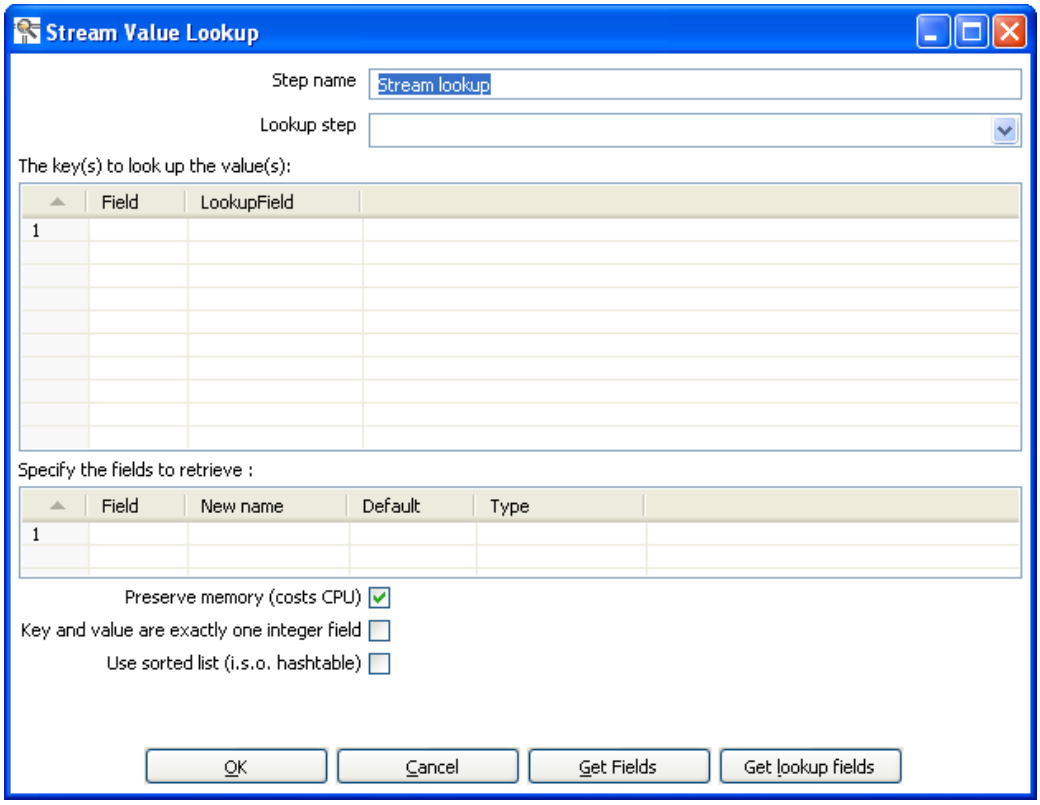
Option	Description
Keys to look up table	Specify the keys necessary to perform the lookup.
Values to return table	Select the fields from the lookup table to add to the output stream.
Do not pass the row if the lookup fails	Check to avoid passing a row when the lookup fails.
Faile on multiple results?	Check this option to cause the step to fail if the lookup returns multiple results.
Order by	The order by field allows you to specify a field and order type (ascending/descending) for how the data is retrieved.
Get Fileds	Click to return a list of available fields from the input stream(s) of the step.
Get lookup fields	Click to return a list of available fields from the lookup table to add to the step's output stream.

-

IMPORTANT NOTE: *if other processes are changing values in the table where you do the lookup, it might be unwise to cache values. However, in all other cases, enabling this option can seriously increase the performance because database lookups are relatively slow. If you find that you can't use the cache, consider launching several copies of this step at the same time. This will keep the database busy via different connections. To see how to do this, please see Launching several copies of a step.*

11.6.20. Stream lookup

Icon



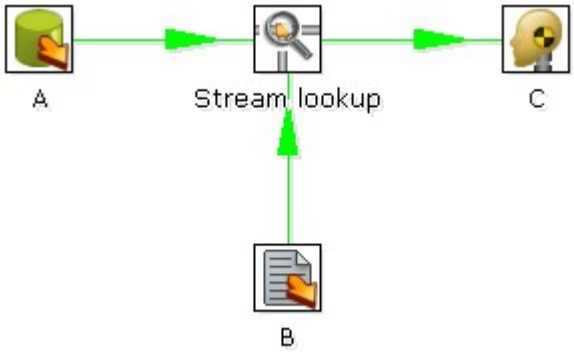
Stream Lookup dialog

11.6.20.1. General description

This step type allows you to look up data using information coming from other steps in the transformation. The data coming from the 'Lookup step' is first read into memory and is then used to look up data from the main stream.

For example, this transformation adds information coming from a text-file (B) to data coming from a database table (A):

The fact that v d by the option: 'Source step' (see below):



Source ste

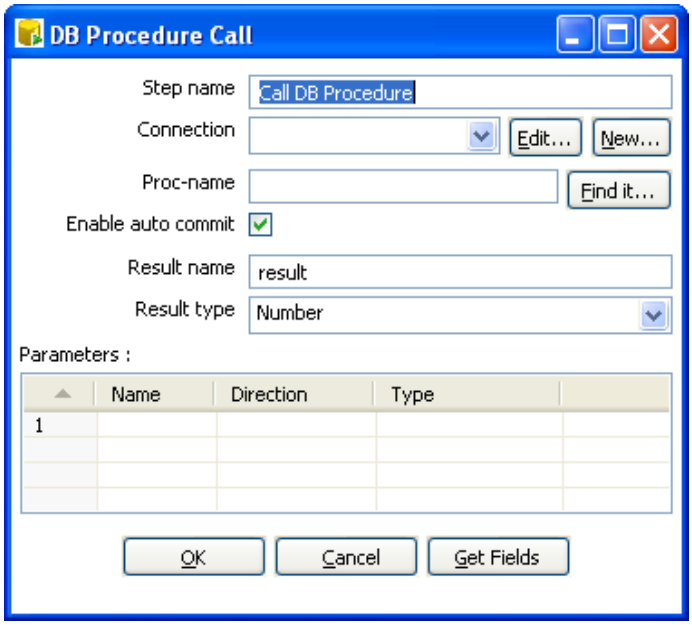
11.6.20.2. Options

The table belo stream lookup:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Lookup step	This is the step name where the lookup data is coming from
The keys to lookup...	Allows you to specify the names of the fields that are used to lookup values. Values are always searched using the "equal" comparison.
Fields to retrieve	You can specify the names of the fields to retrieve here, as well as the default value in case the value was not found or a new fieldname in case you didn't like the old one.
Preserve memory	This will encode rows of data to preserve memory while sorting.
Key and value are exactly one integer field	This will also preserve memory while executing a sort.
Use sorted list	Check this to store values using a sorted list. This provides better memory usage when working with data sets containing wide rows.
Get fields	This will automatically fill in the names of all the available fields on the source side (A). You can then delete all the fields you don't want to use for lookup.
Get lookup fields	This will automatically insert the names of all the available fields on the lookup side (B). You can then delete the fields you don't want to retrieve.

11.6.21. Call DB Procedure

Icon



Call DB Procedure dialog

11.6.21.1. General description

This step type allows you to execute a database procedure (or function) and get the result(s) back.

11.6.21.2. Options

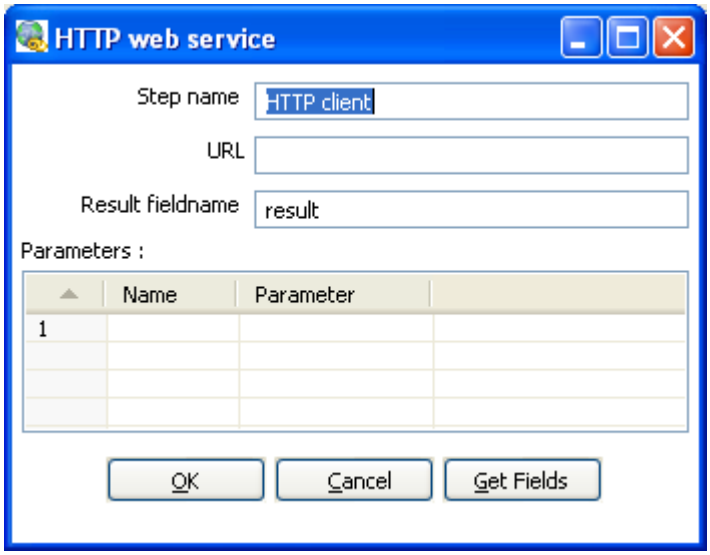
The following table describes the available options for the Call DB Procedure step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Connection	Name of the database connection on which the procedure resides.
Proc-name	Name of the procedure or function to call.
Find it button	Click to search on the specified database connection for available procedures and functions (at the moment only on Oracle and SQLServer).
Enable auto commmit	In some situations you want to do updates in the database using the specified procedure. In that case you can either have the changes done using auto-commit or by disabling this. If auto-commit is disabled, a single commit is being performed after the last row was received by this step.
Result name	Name of the result of the function call, leave this empty in case of procedure.
Result type	Type of the result of the function call. Not used in case of a procedure.
Parameters	List of parameters that the procedure or function needs • Field name: Name of the field. • Direction: Can be either IN (input only), OUT (output only), INOUT (value is changed on the database).

Option	Description
	• Type: Used for output parameters so that Kettle knows what comes back.
Get Fields	This function fills in all the fields in the input streams to make your life easier. Simply delete the lines you don't need and re-order the ones you do need.

11.6.22. HTTP Client

Icon



Call DB Procedure dialog

11.6.22.1. General Description

The HTTP client step performs a very simple call to a base URL with options appended to it like this:

```
http://<URL>?param1=value1&param2=value2&..
```

The result is stored in a String field with the specified name.

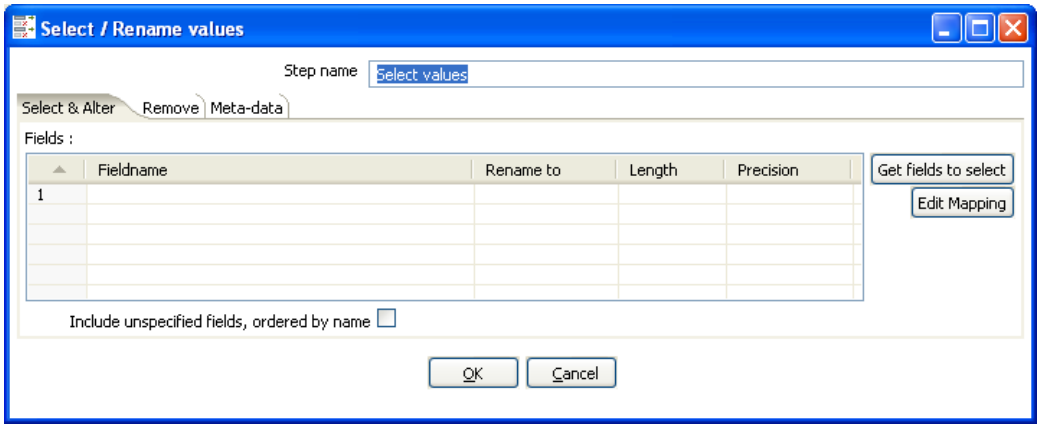
11.6.22.2. Options

The following table describes the options available for the HTTP client step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
URL	The base URL string
Result fieldname	The name of the field to store results
Parameters	This section is where you define the parameter name-value pairs to pass on the URL.

11.6.23. Select values

Icon



Select Values dialog

11.6.23.1. General description

This Select values step is useful for selecting, renaming and configuring the lengh and precesion of the fields on the stream. These operations are organized into different categories:

- Select & Alter - Specify the exact order and name in which the fields have to be placed in the output rows
- Remove - Specify the fields that have to be removed from the output rows
- Meta-data - Change the name, type, length and precision (the meta-data) of one or more fields

11.6.23.2. Select & Alter

The Select & Alter tab provides the following options:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Fields	This tab allows you to rename a field and specify the lenth and precision.
Get fields to select	Click to insert fields from all input steams to the step.
Edit Mapping	Click to open a mapping dialog to easily define mulitple mappings between source and target fields. **Note:** *this step only works if there is only one target output step.*
Include unspecified fields, ordered by name	Enable if you want to implicitly select all other fields from the input stream(s) that are not explicity selected in the Fields section.

11.6.23.3. Remove

This tab allows you to enter the fields that you want removed from th e stream. You can also click the 'Get fields to remove' button to add all fields from the input stream(s). This makes it easier if you are trying to remove several fields. After getting all fields, simply delete any of the fields that you do not want remove from the stream.

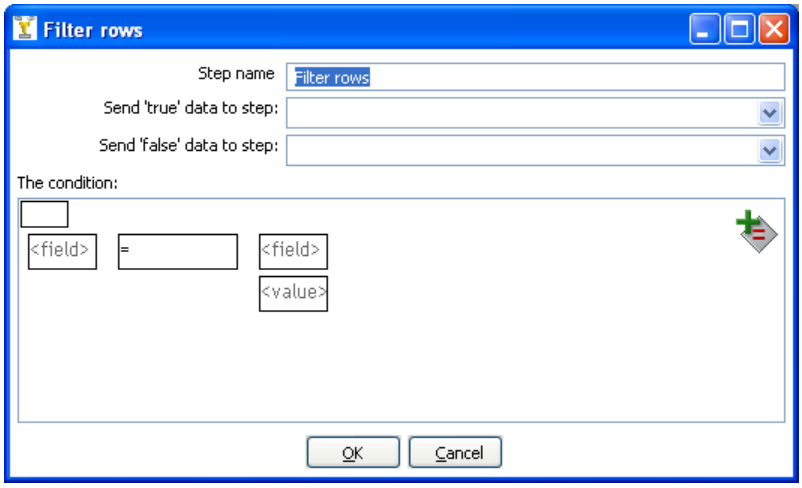
11.6.23.4. Meta-data

This tab allows you to rename, change data types, and change the length and precision of fields coming into the Select Values step. Click the 'Get fields to change' button to add all fields on the input stream(s).

Note: *The type column is useful for cases where you need to set a specific data type to avoid repeated data type conversions. For example, if your transformation is taking advantage of the lazy conversion option and includes a sort step, this could result in repeated data conversions internal to the sort step in order to perform the data comparisons. You can workaround this issue by using the Select Values step to convert your sort key fields to normal data (i.e. from Binary to String).*

11.6.24. Filter rows

Icon



Filter rows dialog

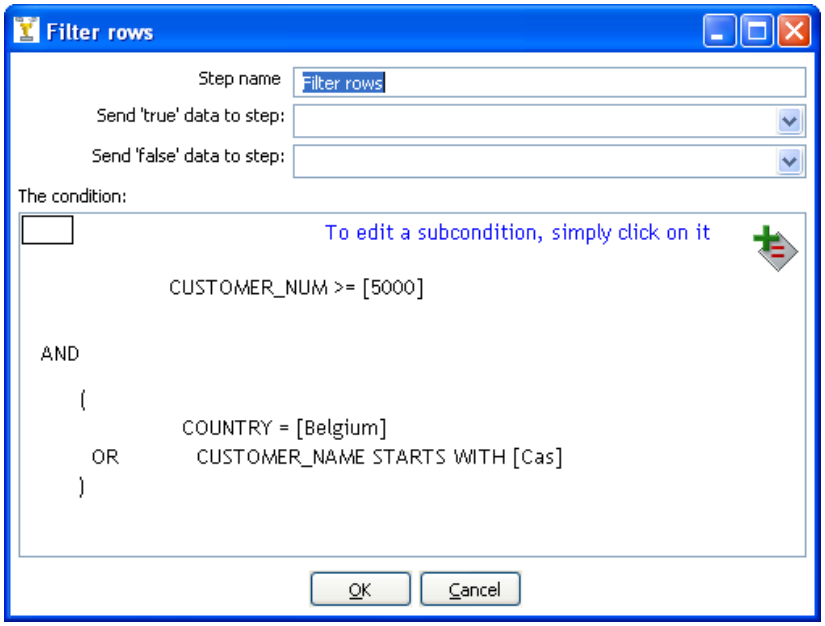
11.6.24.1. General description

This step type allows you to filter rows based upon conditions and comparisons. Once this step is connected to a previous step (one or more and receiving input), you can simply click on the "<field>", "=" and "<value>" areas to construct a condition.

You can add more conditions by clicking on the 'Add condition' icon seen here:

It will convert the original condition to a subcondition and add one more. A subcondition can be edited simply by clicking on it (going down one level into the condition tree).

For example, this is a more complex example:

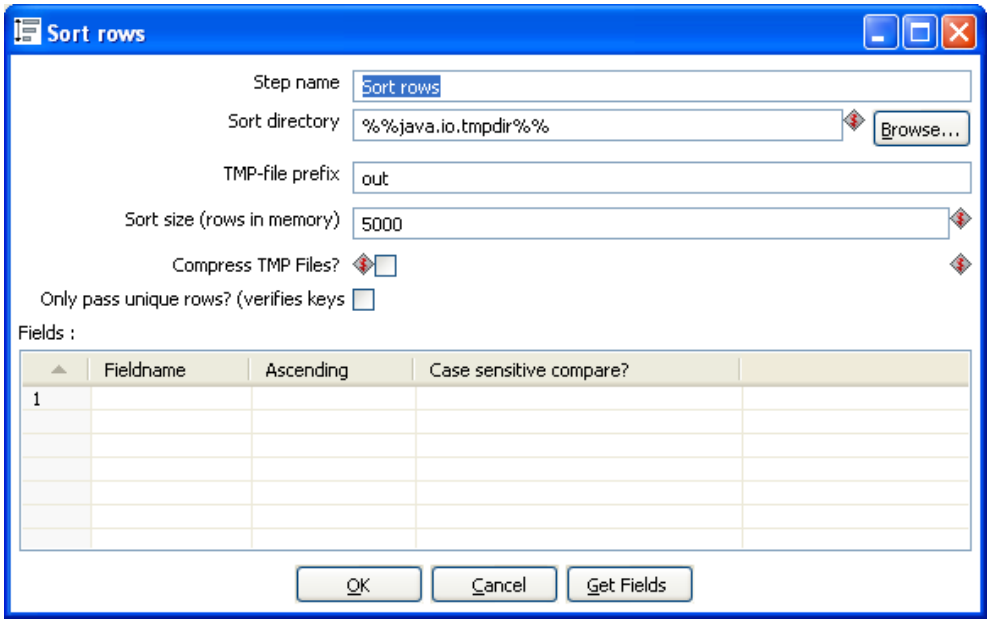


11.6.24.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Send 'true' data to step	The rows for which the condition specified evaluates to true are send to this step.
Send 'false' data to step	The rows for which the condition specified evaluates to false are send to this step.
The Condition	Click the 'NOT' button in the upper left to negate the condition. Click on the <Field> buttons to select from a list of fields from the input stream(s) to build your condidion(s). Click on the <value> button to enter a specific value into your condition(s). To delete a condition, right-click on it and select 'Delete Condition'.
Add Condition button	Click to add a condition.

11.6.25. Sort rows

Icon



Sort rows dialog

11.6.25.1. General description

This step type sorts rows based upon the fields you specify and whether or not they should be sorted ascending or descending.

NOTE: *Kettle has to sort rows using temporary files when the number of rows exceeds 5000.*

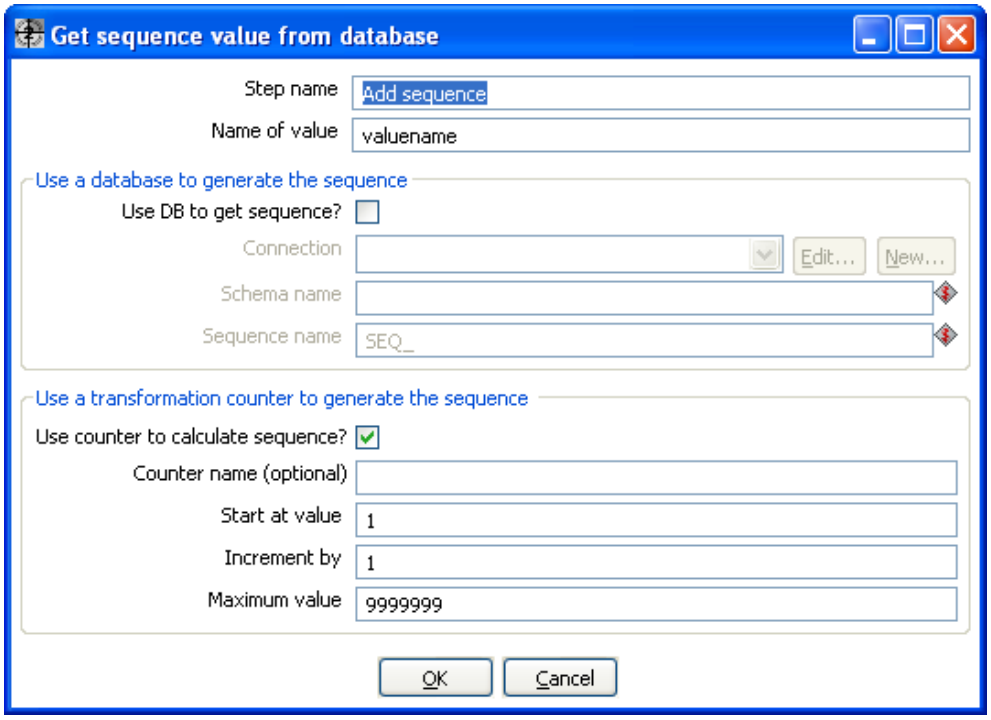
11.6.25.2. Options

The following table describes the options for the Sort step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Sort directory	This is the directory in which the temporary files are stored in case it is needed. The default is the standard temporary directory for the system.
TMP-file prefix	Choose a recognizable prefix in order to recognize the files when they show up in the temp directory.
Sort size	The more rows you can store in memory, the faster the sort gets. This is because less temporary files need to be used and less I/O is generated.
Compress TMP Files	This option compresses temporary files when they are needed to complete the sort.
Only pass unique rows?	Enable this option if you want to only pass unique rows to the output stream(s).
Fields table	Specify the fields and direction (ascending/decending) to sort. You can optionally specify whether or not to perform a case sensitive sort.
Get Fields button	Click to retrieve a list of all fields coming in on the stream(s).

11.6.26. Add sequence

Icon



Sort rows dialog

11.6.26.1. General description

This step will add a sequence to the stream. A sequence is an ever-changing integer value with a certain start and increment value. You can either use a database (Oracle) sequence to determine the value of the sequence, or have it generated by Kettle..

NOTE*: Kettle sequences are only unique when used in the same transformation. Also, they are not stored, so the values start back at the same value every time the transformation is launched.*

11.6.26.2. Options

The following table describes the options for the Add Sequence step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Name of value	Name of the new sequence value that is added to the stream.
Use DB to generate the sequence	Enable this option if you want the sequence to be driven by a database sequence. • **Connection name**: choose the name of the connection on which the database sequence resides. • **Schema name**: optionally specify the table's schema name • **Sequence name**: allows you to enter the name of the database sequence.
Use a transformation	Enable this option if you want the sequence to be generated by

Option	Description
counter to generate the sequence	Kettle. • **Use counter to calculate sequence**: Enable this option if you want the sequence to be generated by Kettle. • **Counter name (optional)**: if multiple steps in a transformation generate the same value name, this option would allow you to specify the name of the counter you want to associate the sequence with. This would avoid forcing unique sequencing across multiple steps. Attention: *In this case you have to ensure that: start, increment and maximum value of all counters with the same name are identical, otherwise the result is unpredictable.* • **Start at**: give the start value of the sequence. • **Increment by**: give the increment of the sequence. • **Maximum value**: this is the maximum value after which the sequence will start back at the start value (Start At). **Examples:** `Start at = 1, increment by = 1, max value = 3` `→ This will produce: 1, 2, 3, 1, 2, 3, 1, 2…` `Start at = 0, increment by = -1, max value = -2` `→ This will produce: 0, -1, -2, 0, -1, -2, 0…`

11.6.27. Dummy (do nothing)

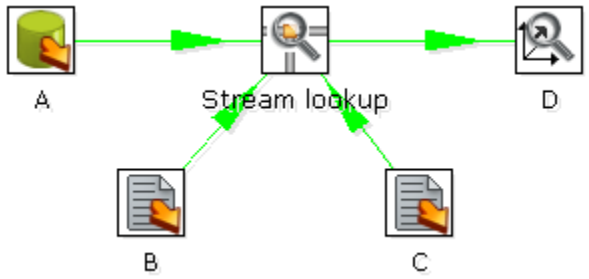
Icon

Dummy dialog

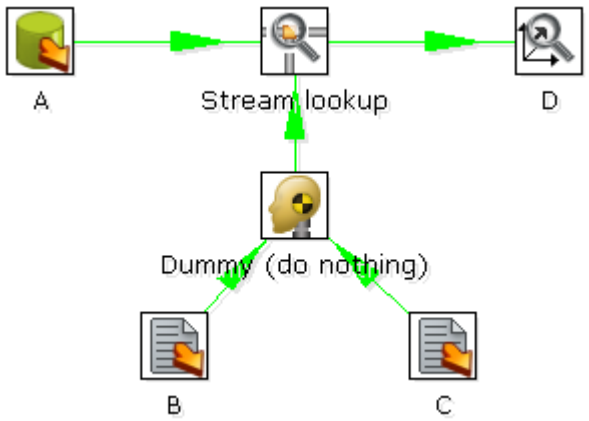
11.6.27.1. General description

This step does not do anything. Its main function is perform as a placeholder in case you want to test something. For example, to have a transformation, you need at least 2 steps connected to each other. If you want to test for example a test file input step, you can connect it to a dummy step.

Here is another example using the Dummy step, starting with the following transformation:

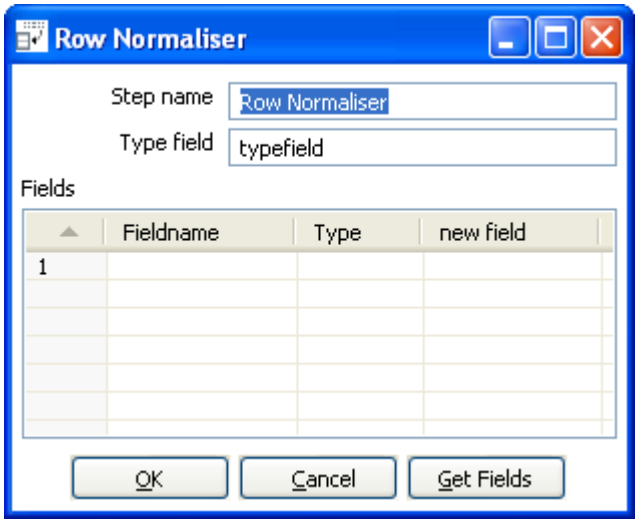


Unfortunately, the 'Stream Lookup' step can only read lookup information from one stream. The Dummy step can be used to work around this limitation like this:



11.6.28. Row Normaliser

Icon



Row Normalser dialog

11.6.28.1. General description

This step normalizes data back from pivoted tables.

For example, starting with this example of product sales data:

Month	Product A	Product B	Product C
2003/01	10	5	17
2003/02	12	7	19
...	...	...	...

The Row Normaizer step will convert this data into the following format so that it is easier to update your fact table:

Month	Product	sales
2003/01	A	10
2003/01	B	5
2003/01	C	17
2003/02	A	12
2003/02	B	7
2003/02	C	19
...	...	...

11.6.28.2. Options

The following options are available for the Row Normaliser Step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Typefield	The name of the type field. (Product in our example)
Fields table	This is a list of the fields you want to normalizer, you will need to set

Option	Description
	the following properties for each selected field: • **Fieldname**: Name of the fields to normalize (Product A ☐ C in our example). • **Type**: Give a string to classify the field (A, B or C in our example). • **New field**: You can give one or more fields where the new value should transferred to (sales in our example).
Get Fields button	Click to retrieve a list of all fields coming in on the stream(s).

11.6.28.3. Example – normalising multiple rows in a single step

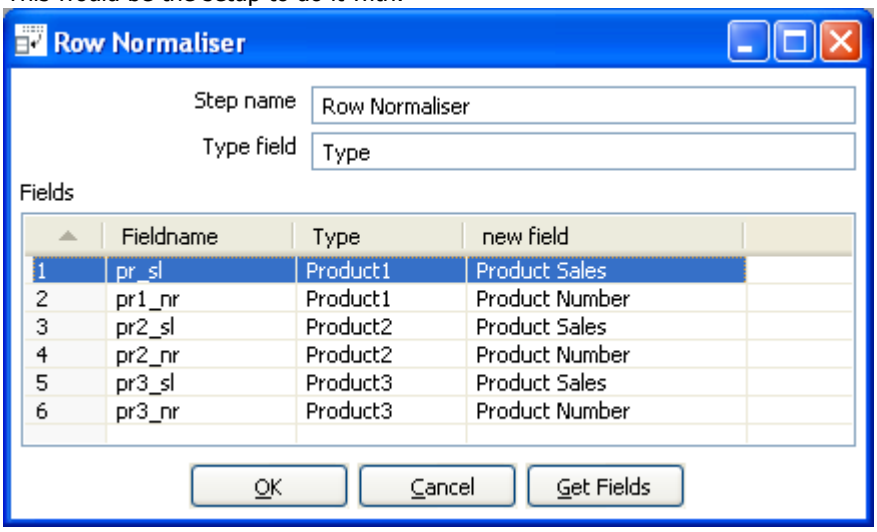
The following example illustrates using the Row Normalaser step to normalize more than one row at a time. Beginning with the following data format:

DATE	PR1_NR	PR1_SL	PR2_NR	PR2_SL	PR3_NR	PR3_SL
20030101	5	100	10	250	4	150
…	…	…	…	…	…	…

You can convert this into:

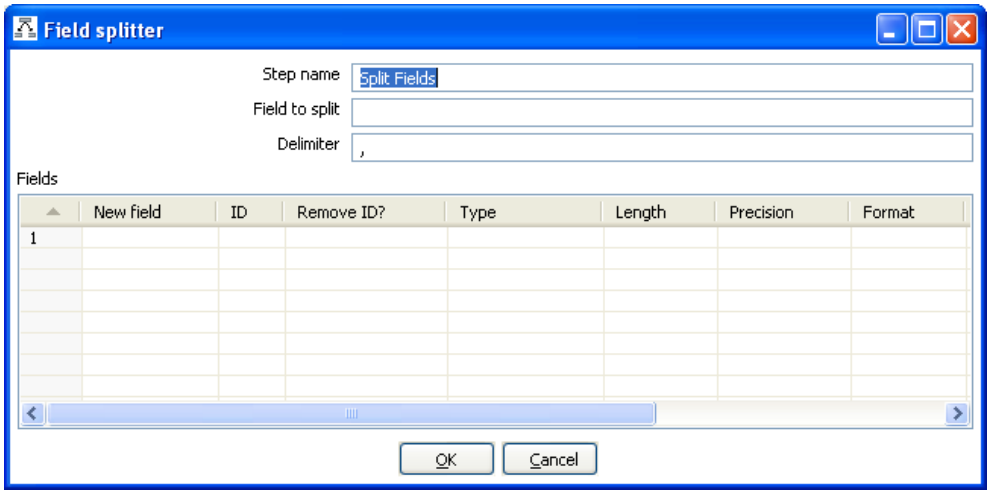
DATE Type	Product	Sales	Product Number
20030101	Product1	100	5
20030101	Product2	250	10
20030101	Product3	150	4
…	…	…	…

This would be the setup to do it with:



11.6.29. Split Fields

Icon



Split Fields dialog

11.6.29.1. General description

This step allows you to split fields based upon delimiter information.

11.6.29.2. Options

The following options are available for configuring the Split Fields step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Field to split	The name of the field you want to split.
Delimiter	Delimiter that determines the end of a field.
Fields table	This table is where you define the properties for each new field created by the split. For each new field, you will need to define the field name, data type and other properties.

11.6.29.3. Split fields examples

Example 1:

SALES_VALUES field containing: "500,300,200,100"

Use these settings to split the field into 4 new fields:

- Delimiter: ,
- Field: SALES1, SALES2, SALES3, SALES4
- Id:
- remove ID no, no, no, no
- type: Number, Number, Number, Number
- format: ###.##, ###.##, ###.##, ###.##
- group:
- decimal: .

- currency:
- length: 3, 3, 3, 3
- precision: 0, 0, 0, 0

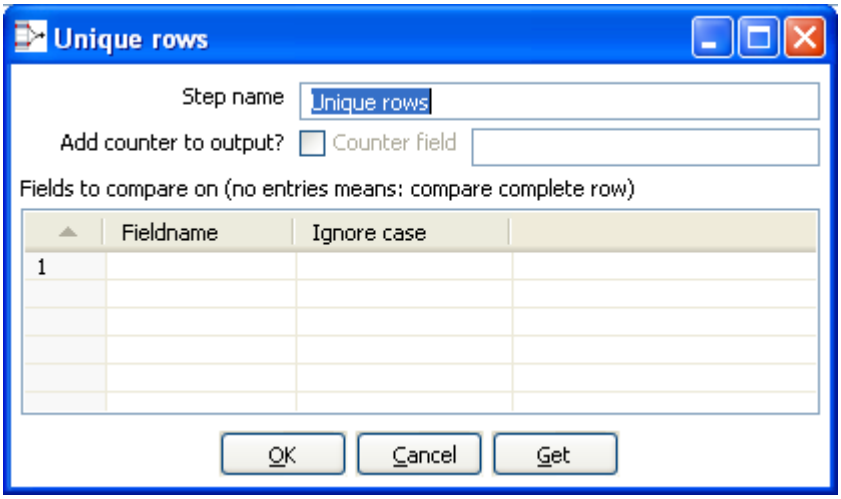
Example 2:
SALES_VALUES field containing "Sales2=310.50, Sales4=150.23"

Use these settings to split the field into 4 new fields:

- Delimiter: ,
- Field: SALES1, SALES2, SALES3, SALES4
- Id: Sales1=, Sales2=, Sales3=, Sales4=
- remove ID yes, yes, yes, yes
- type: Number, Number, Number, Number
- format: ###.##, ###.##, ###.##, ###.##
- group:
- decimal: .
- currency:
- length: 7, 7, 7, 7
- precision: 2, 2, 2, 2

11.6.30. Unique rows

Icon



Unique rows dialog

11.6.30.1. General description

This step removes duplicate rows from the input stream(s).

IMPORTANT NOTE: *Make sure that the input stream is sorted! Otherwise only consecutive double rows are evaluated correctly.*

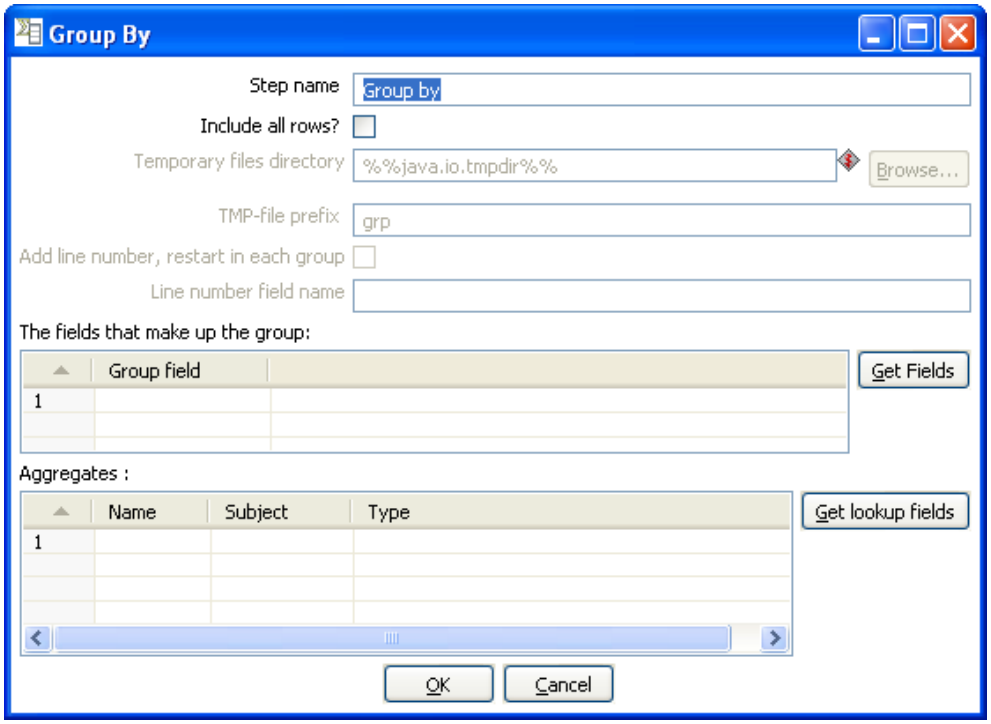
11.6.30.2. Options

The following table describes all options for the Unique rows step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Add counter to output?	
Fields to compare table	Specify the field names you want to force uniqueness on or click the 'Get' button to insert all fields from the input stream(s). You can (optionally) choose to ignore case by setting the 'Ignore case' flag to Y. For example: Kettle, KETTLE, kettle all will be the same in case the compare is done case insensitive. In this case, the first occurrence (Kettle) will be passed to the next step(s).

11.6.31. Group By

Icon



Group by dialog

11.6.31.1. General description

This step allows you to calculate values over a defined group of fields. Examples of common use cases are:

- calculate the average sales per product
- get the number of yellow shirts that we have in stock

11.6.31.2. Options

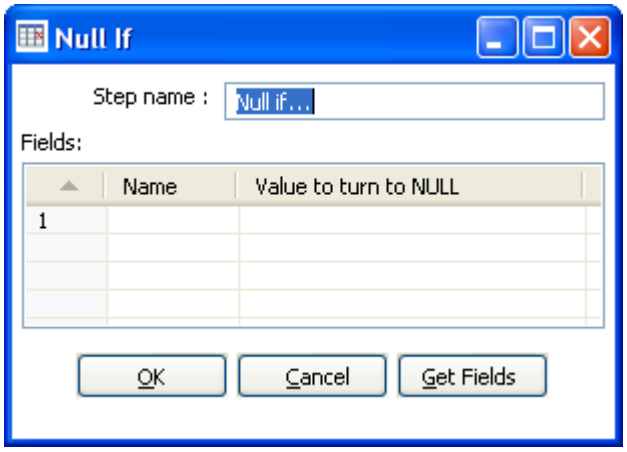
The following table provides a description of the options available for the Group By step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Include all rows?	Check this if you want all rows in the output, not just the aggregation. To differentiate between the 2 types of rows in the output, we need a flag in the output. You need to specify the name of the flag field in that case. (the type is boolean)
Temporary files directory	This is the directory in which the temporary files are stored in case it is needed. The default is the standard temporary directory for the system.
TMP-file prefix	Specify the file prefix used when naming temporary files.
Add line number, restart in each group	Check this if you want to add a line number that restarts at 1 in each group.
Line number field name	Check this if you want to add a line number that restarts at 1 in each

Option	Description
	group.
Group fields table	Specify the fields over which you want to group. You can click the 'Get Fields' button to add all fields from the input stream(s).
Aggregates table	Specify the fields that need to be aggregated, the method and the name of the resulting new field.

11.6.32. Null If

Icon



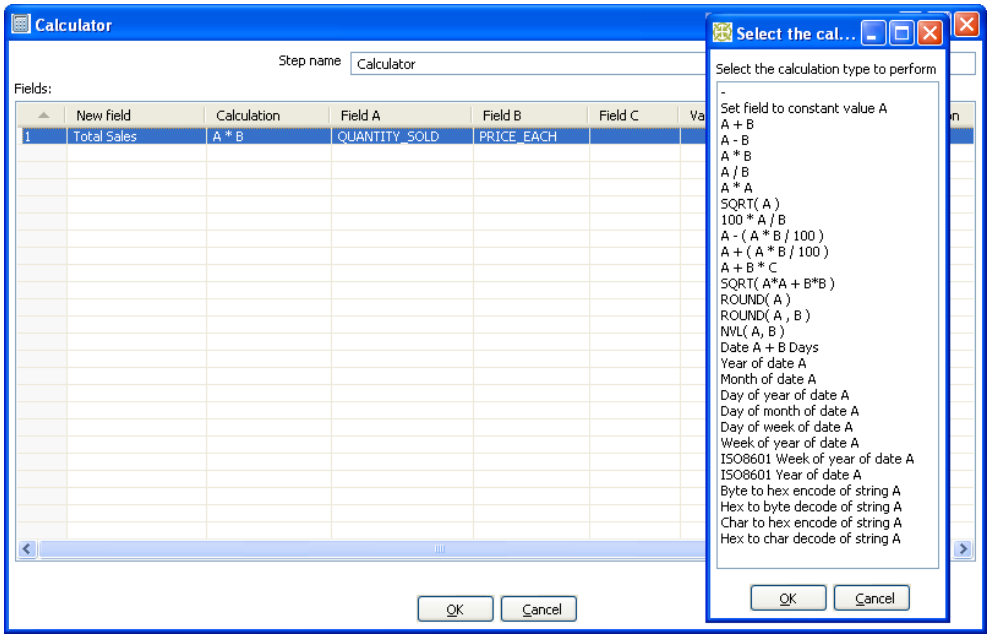
Null If dialog

11.6.32.1. General description

If the string representation of a certain field is equal to the specified value, then the value is set the null (empty). You can add all fields from the input stream(s) using the 'Get Fields'.

11.6.33. Calculator

Icon



Calculator dialog

11.6.33.1. General description

This calculator step provides a pre-defined functions that can be executed on input field values. If you have a need for other generic, often used functions, please visit our community page and let us know about your enhancement request.

Note: *An important advantage Calculator has over custom JavaScript scripts is that the execution speed of Calculator is many times that of a script.*

Besides the arguments (Field A, Field B and Field C) you also need to specify the return type of the function. You can also opt to remove the field from the result (output) after all values were calculated. This is useful for removing temporary values.

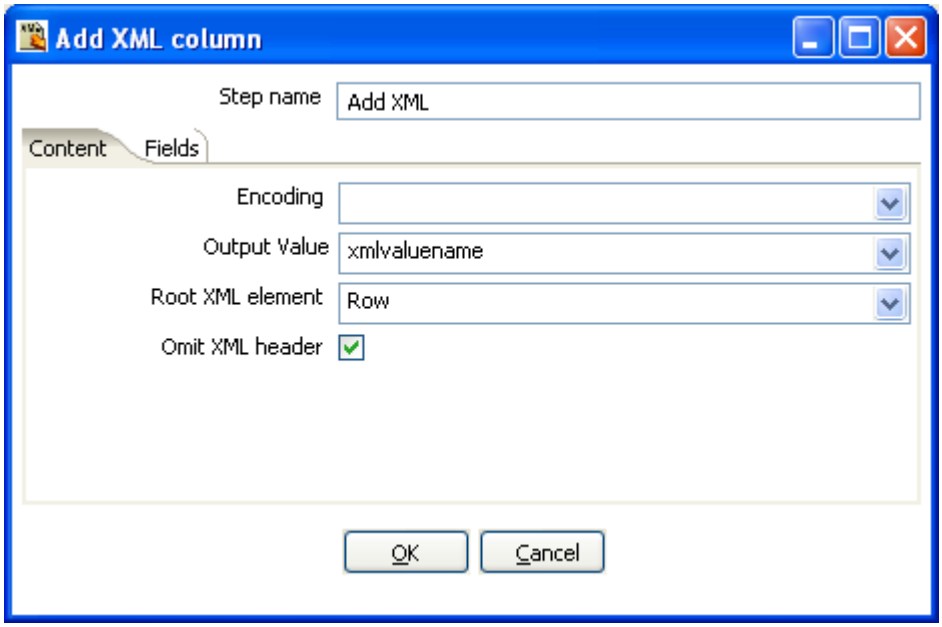
11.6.33.2. 10.4.34.4 Function List

Function	Description	Required fields
Set field to constant A	Create a field with a constant value	A
A + B	A plus B	A and B
A - B	A minus B	A and B
A * B	A multiplied by B	A and B
A / B	A divided by B	A and B
A * A	The square of A	A
SQRT(A)	The square root of A	A
100 * A / B	Percentage of A in B	A and B
A - (A * B / 100)	Subtract B% of A	A and B
A + (A * B / 100)	Add B% to A	A and B

Function	Description	Required fields
A + B *C	Add A and B times C	A, B and C
SQRT(A*A + B*B)	Calculate √(A2+B2)	A and B
ROUND(A)	Round A to the nearest integer	A
ROUND(A, B)	Round A to B decimal positions	A and B
Set field to constant A	Create a field with a constant value	A
Date A + B days	Add B days to Date field A	A and B
Year of date A	Calculate the year of date A	A
Month of date A	Calculate number the month of date A	A
Day of year of date	A Calculate the day of year (1-365)	A
Day of month of date A	Calculate the day of month (1-31)	A
Day of week of date A	Calculate the day of week (1-7)	A
Week of year of date A	Calculate the week of year (1-54)	A
ISO8601 Week of year of date A	Calculate the week of the year ISO8601 style (1-53)	A
ISO8601 Year of date A	Calculate the year ISO8601 style	A
Byte to hex encode of string A	Encode bytes in a string to a hexadecimal representation	A
Hex encode of string A	Encode a string in its own hexadecimal representation	A
Char to hex encode of string A	Encode characters in a string to a hexadecimal representation	A
Hex decode of string A	Decode a string from its hexadecimal representation (add a leading 0 when A is of odd length)	A

11.6.34. XML Add

Icon



Add XML dialog

11.6.34.1. General description

This step allows you to encode the content of a number of fields in a row in XML. This XML is added to the row in the form of a String field.

11.6.34.2. Content Tab

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Encoding	The encoding to use. This encoding is specified in the header of the XML file.
Output Value	The name of the new field that will contain the XML
Root XML element	The name of the root element in the generated XML.
Omit XML header	Check to not include the XML header in the output.

11.6.34.3. Fields

The Fields tab is where you configure the output fields and their formats. The table below describes each of the available properties for a field:

Option	Description
Fieldname	Name of the field.
Element name	The name of the element in the XML file to use.
Type	Type of the field can be either String, Date, or Number.
Format	Format mask to convert with: see Number Formats for a complete description of format specifiers.
Length	Output string is padded to this length if it is specified.
Precesion	The precision to use.

Option	Description
Currency	Symbol used to represent currencies like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10,000.00) or "," (5.000,00)
Grouping	A grouping can be a "," (10,000.00) or "." (5.000,00)
Null	The string to use in case the field value is null. Attribute: make this an attribute (N means : element)

11.6.34.4. Add XMLExample

Use Case

I have data that comes in a variety of classes and I would like to store it as XML in my database. For example, I want to turn the raw data into the database layout below:

Raw data

SHAPE	COLOUR	id	X	Y	RADIUS
circle	blue	1	3	5	5
circle	red	2	1	3	5
circle	blue	5	5	9	5
circle	blue	6	8	2	5
circle	red	7	9	7	5

SHAPE	COLOUR	id	X	Y	LENGTH	WIDTH
rectangle	blue	3	3	1	6	4
rectangle	red	6	2	4	6	4
rectangle	blue	10	8	2	6	4
rectangle	red	12	7	8	6	4
rectangle	Blue	14	5	2	6	4

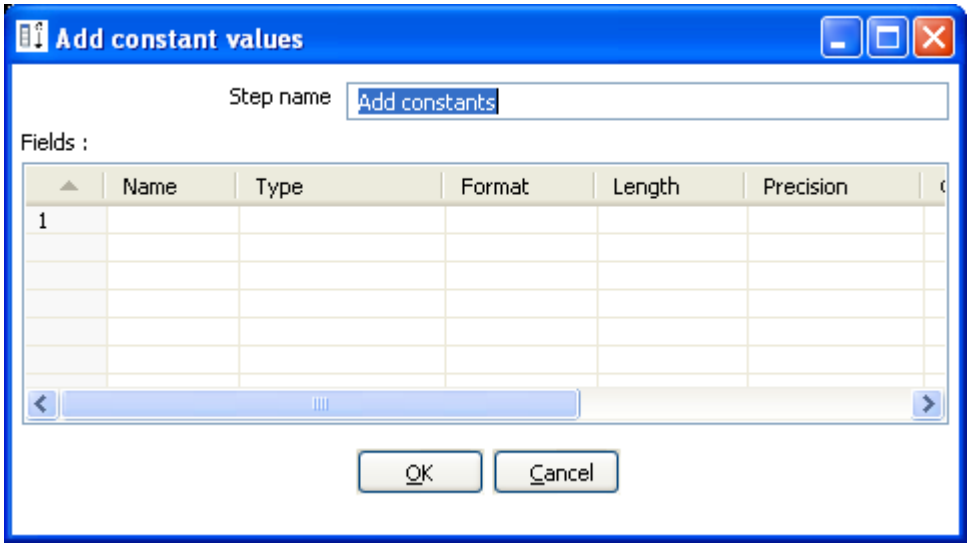
Output Sample

ID	X	Y	CLASS_DATA
3	4	7	<SHAPE type="circle"> <COLOUR>blue</COLOUR> <RADIUS> 5</RADIUS </SHAPE>
1	6	3	<SHAPE type="rectangle"> <COLOUR>blue</COLOUR> <WIDTH> 4</WIDTH> <LENGTH> 6</LENGTH> </SHAPE>
2	8	8	<SHAPE type="rectangle"> <COLOUR>blue</COLOUR> <WIDTH> 4</WIDTH> <LENGTH>6</LENGTH> </SHAPE>
5	5	2	<SHAPE type="circle">

```
    <COLOUR>blue</COLOUR>
    <RADIUS> 5</RADIUS>
</SHAPE>
```

11.6.35. Add constants

Icon



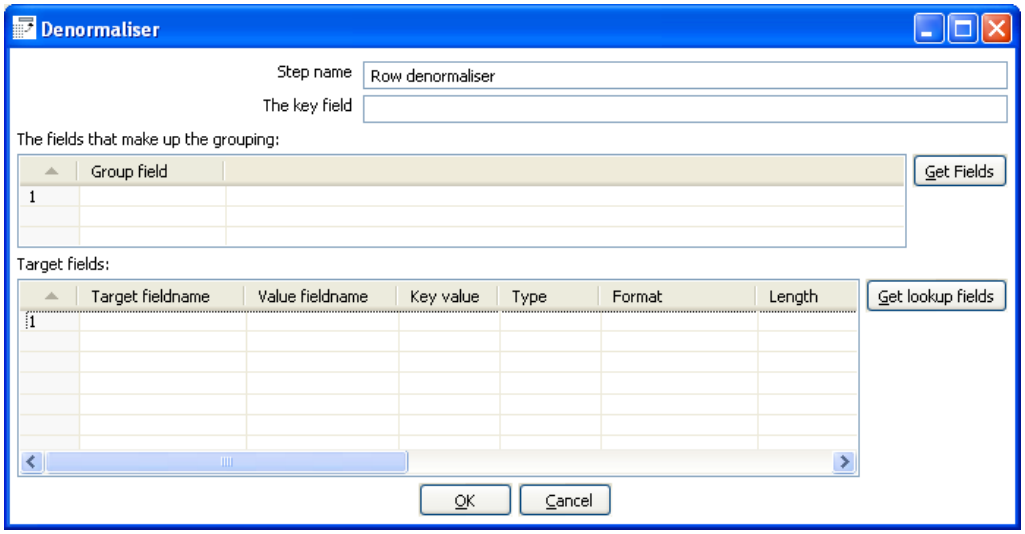
Add constants dialog

11.6.35.1. General description and usage

The Add contant values step is a simple and fast performing way to add constant values to the stream. To add a constant, simply specify the name, type and value in the form of a string. Then, specify the formats to convert the value into the chosen data type.

11.6.36. Row Denormaliser

Icon



Denormaliser dialog

11.6.36.1. General description

This step allows you de-normalize data by looking up key-value pairs. It also allows you to immediately convert data types.

11.6.36.2. Options

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
Key field	The field that defined the key.
Group fields	Specify the fields that make up the grouping here.
Target fields	Specify the fields to de-normalize. You do it by specifying the String value for the key field (see above). Options are provided to convert data types. Mostly people use Strings as key-value pairs so you often need to convert to Integer, Number or Date. In case you get key-value pair collisions (key is not unique for the group specified) you can specify the aggregation method to use.

11.6.37. Flattener

Icon

Flattener dialog

11.6.37.1. General description

This step allows you flatten sequentially provided data.

11.6.37.2. Options

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
The field to flatten	The field that needs to be flattened into different target fields.
Target fields	The name of the target field to flatten to.

11.6.37.3. Flattener Example

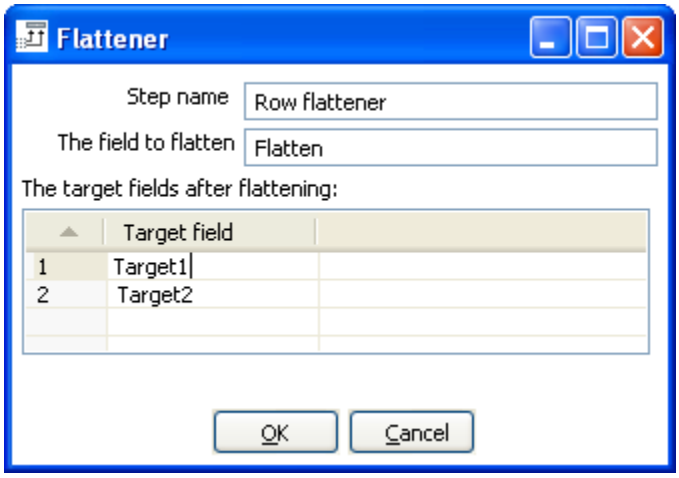
Beginning with the following data set:

Field1	Field2	Field3	Flatten
A	B	C	One
A	B	C	Two
D	E	F	Three
D	E	F	Four

This can be flattened to:

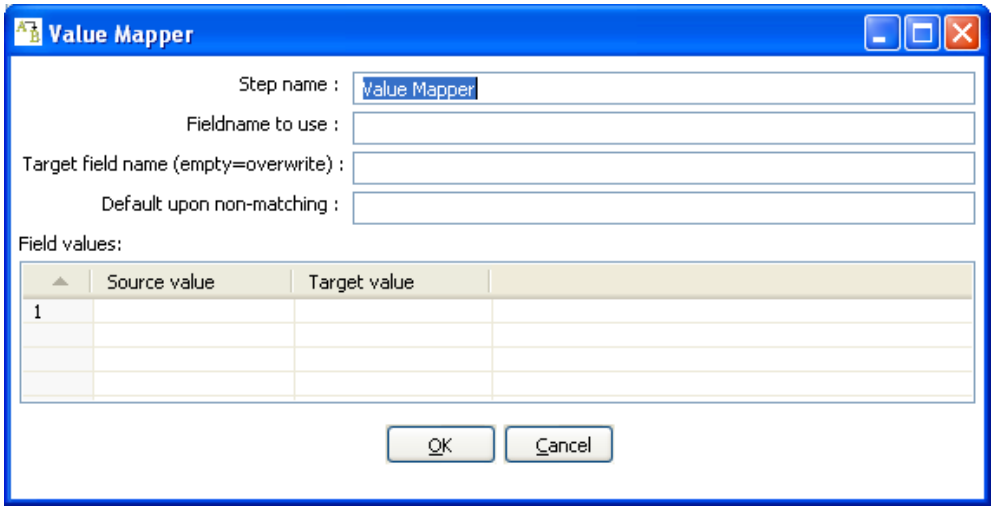
Field1	Field2	Field3	Target1	Target2
A	B	C	One	Two
D	E	F	Three	Four

In the example above, this is what the dialog looks like:



11.6.38. Value Mapper

Icon



Value Mapper dialog

11.6.38.1. General description

This step maps string values from one value to another. Usually, you will want to solve this problem by storing the conversion table in a database. However, this step provides a simple alternative.

For example, if you want to replace language codes:

Fieldname to use: LanguageCode

Target fieldname: LanguageDesc

Source/Target: EN/English, FR/French, NL/Dutch, ES/Spanish, DE/German, ...

NOTE: *It is also possible to convert a null field or empty String value to a non-empty value. Leave the "Source value" field empty for this. It is obviously only possible to specify one of these empty source field values.*

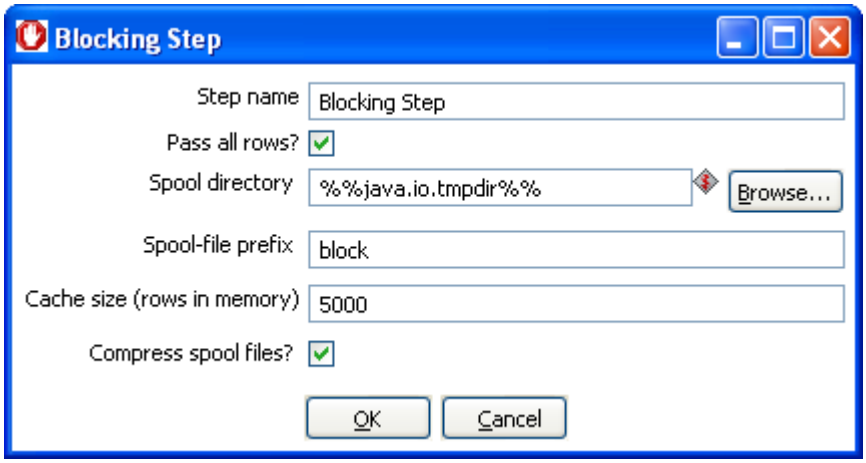
11.6.38.2. Options

The following properties are used to define the mappings:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Fieldname to use	Field to use as the mapping source.
Target field name	Field to use as the mapping Target
Default upon non-matching	Defines a default value for situations where the source value is not empty, but there's no match.
Field values table	Contans the mapping of source value to converted target value.

11.6.39. Blocking step

Icon



Blocking dialog

11.6.39.1. General description

This simple step blocks all output until the very last row was received from the previous step. At that time either the last row is then sent off to the next step or the complete input is sent of to the next step. This step then knows that all previous steps have finished. You can use this for triggering plugins, stored procedures, java scripts, ... or for synchronization purposes.

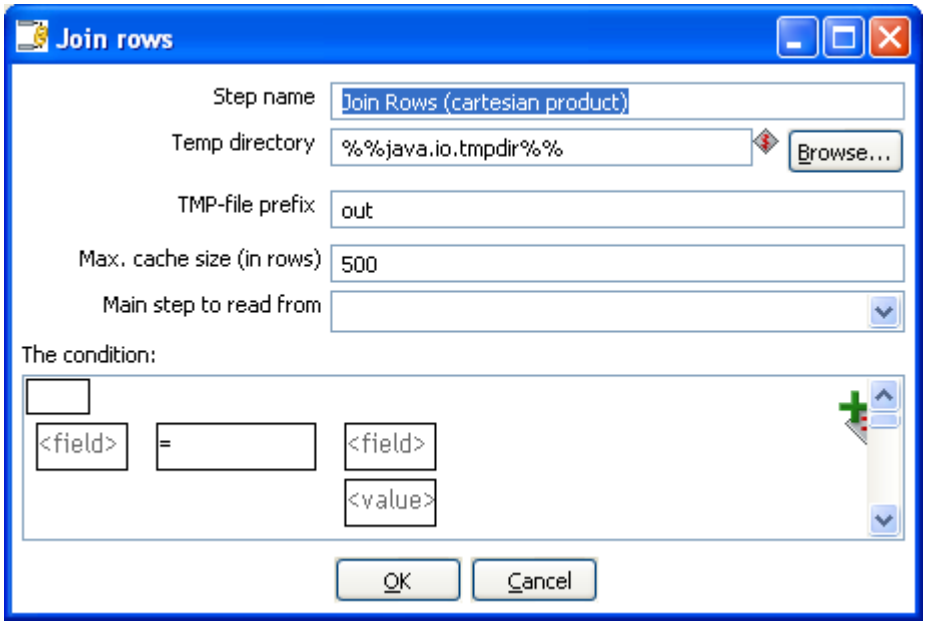
11.6.39.2. Options

The following table describes the options for the Blocking step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Pass all rows?	Determines whether to pass 1 rows or all rows.
Spool directory	This is the directory in which the temporary files are stored in case it is needed. The default is the standard temporary directory for the system.
Spool-file prefix	Choose a recognizable prefix in order to recognize the files when they show up in the temp directory.
Cache size	The more rows you can store in memory, the faster the step works.
Compress spool files?	This option compresses temporary files when they are needed.

11.6.40. Join Rows (Cartesian product)

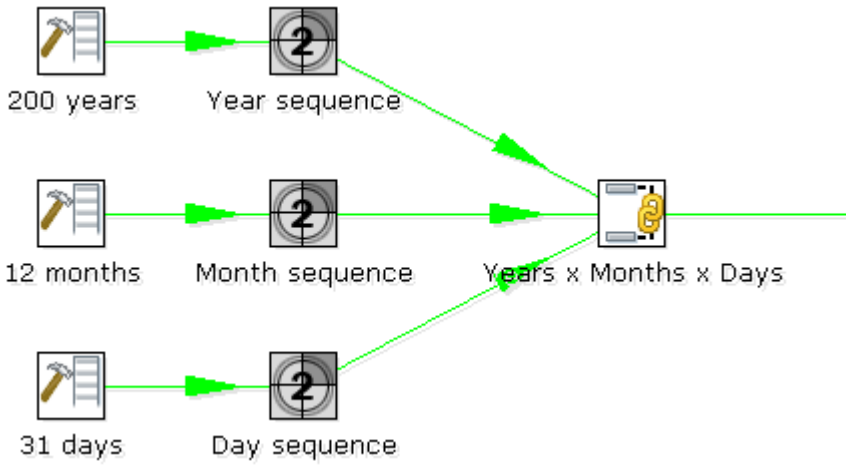
Icon



Join rows dialog

11.6.40.1. General description

This step allows you to produce combinations (Cartesian product) of all rows on the input streams. This is an example:



The "Years x Months x Days" step outputs all combinations of Year, Month and Day. (1900, 1, 1 → 2100, 12, 31) and can be used to create a date dimension.

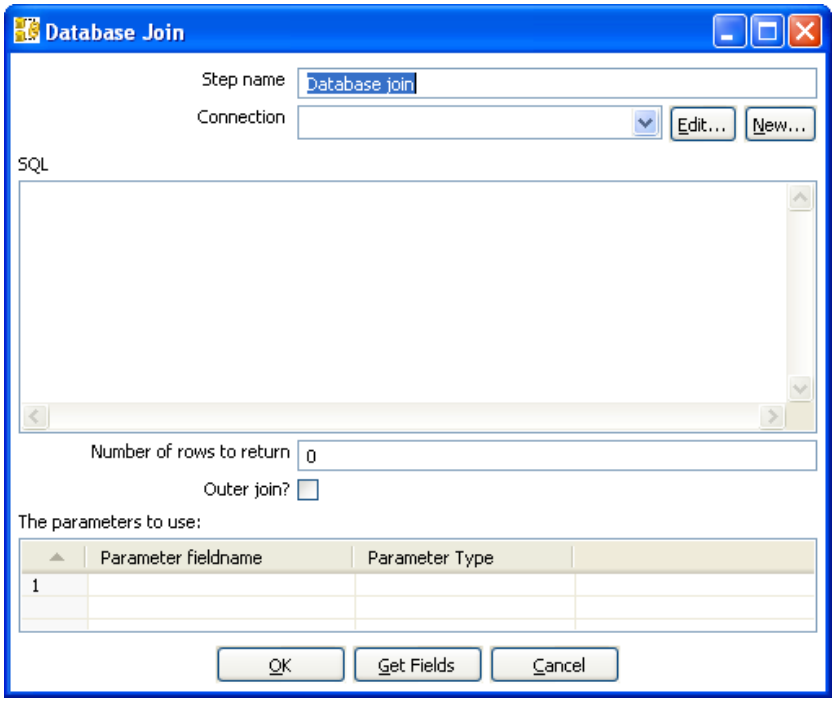
11.6.40.2. Options

The following table describes the options for configuring the Join rows step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Temp directory	Specify the name of the directory where the system stores temporary files in case you want to combine more then the cached number of rows.
TMP-file prefix	This is the prefix of the temporary files that will be generated.
Max. cache size	The number of rows to cache before the systems reads data from temporary files. This is needed in case you want to combine large row sets that don't fit into memory.
Main step to read from	Specifies the step to read the most data from. This step is not cached or spooled to disk, the others are.
The Condition(s)	You can enter a complex condition to limit the number of output rows.

11.6.41. Database Join

Icon



Database Join dialog

11.6.41.1. General description

Using data from previous steps, this step allows you to run a query against a database.

The parameters for this query can be specified:

as question marks (?) in the SQL query.

- as fields in the data grid.

The two need to be in the same order. For example, this step allows you to run queries looking up the oldest person that bought a certain product like this:

```
SELECT      customernr
FROM        product_orders, customer
WHERE       orders.customernr = customer.customernr
AND         orders.productnr = ?
ORDER BY    customer.date_of_birth
```

All you then need to specify as a parameter is the productnr and you'll get the customernr included in the result.

11.6.41.2. Options

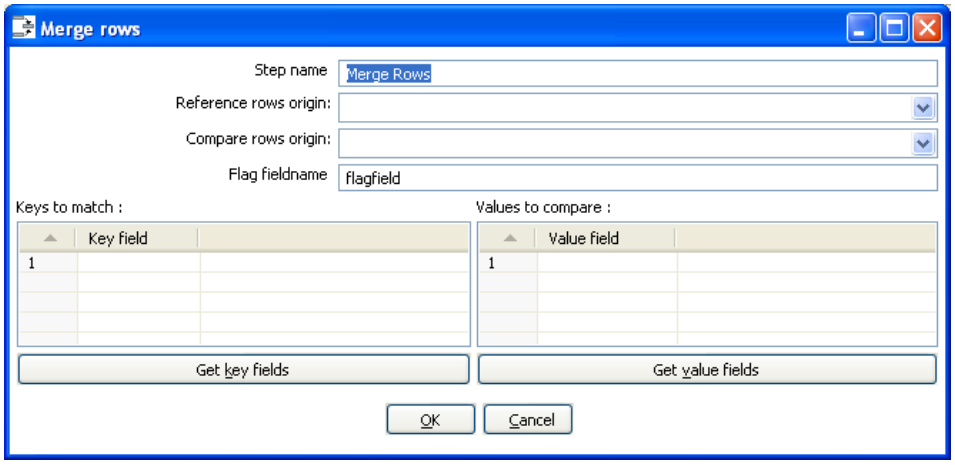
The following table desribes the options for the Database Join step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Connection	The database connection to use.

Option	Description
SQL	SQL query to launch towards the database, use question marks as parameter placeholders.
Number of rows to return	0 means all, any other number limits the number of rows.
Outer join?	Check this to always return a result, even if the query didn't return a result.
Parameters table	Specify the fieldns containg parameters and the parameter type.

11.6.42. Merge rows

Icon



Merge rows dialog

11.6.42.1. General description

This step simply allows you to compare two streams of rows. This is useful if you want compare data from two different times. It is often used in situations where the source system of a data warehouse does not contain a date of last update.

The two streams of rows, a reference stream (the old data) and a compare stream (the new data), are merged. Each time only the last version of a row is passed onto the next steps. The row is marked as follows:

- "identical" - The key was found in both streams and the values to compare were identical;
- "changed" - The key was found in both streams but one or more values is different;
- "new" - The key was not found in the reference stream;
- "deleted" - The key was not found in the compare stream.

The row coming from the compare stream is passed on to the next steps, except for the "deleted" case.

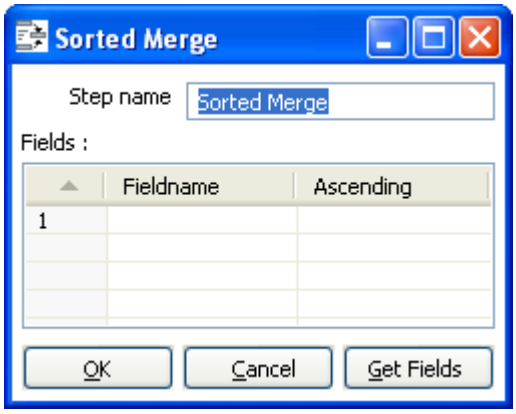
IMPORTANT: *both streams need to be sorted on the specified key(s).*

11.6.42.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Reference rows origin	Specify the step orgin for the reference rows.
Compare rows origin	Specify the step origin for the compare rows.
Flag fieldname	Specify the name of the flag flag field on the output stream.
Keys to match	Specify fields containing the keys to match on. Click the 'Get key fields' button to insert all of the fields originating from the reference rows step.
Values to compare	Specify fields contaning the values to compare. Click the 'Get value fields' button to insert all of the fields from the originating value rows step.

11.6.43. Sorted Merge

Icon



Sorted Merge dialog

11.6.43.1. General description and usage

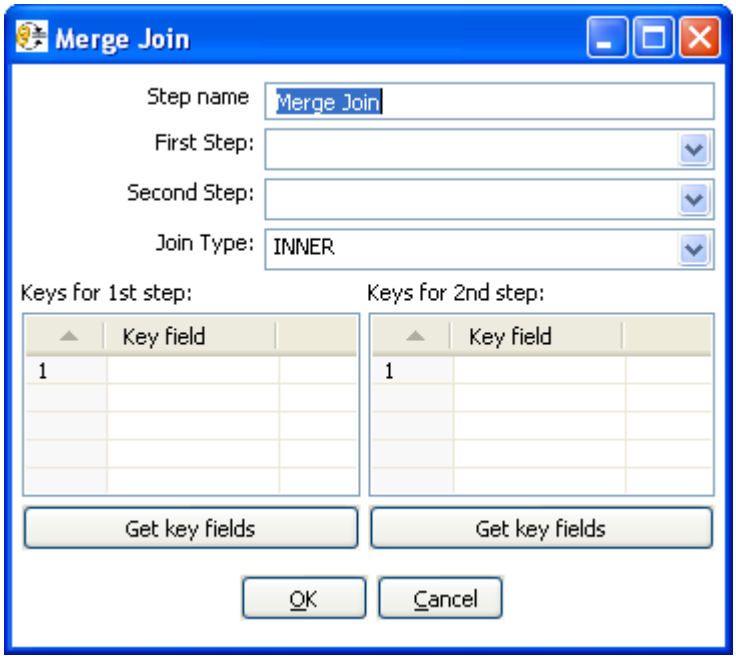
The Sorted Merge step merges rows coming from multiple input steps providing these rows are sorted themselves on the given key fields.

11.6.43.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Fields table	Specify the fieldname and sort direction (ascending/decending). Click the 'Get Fields' button to retrieve a list of fields from the input stream(s).

11.6.44. Merge Join

Icon



Merge Join dialog

11.6.44.1. General description and usage

The Merge Join step performs a classic merge join between data sets with data coming from two different input steps. Join options include INNER, LEFT OUTER, RIGHT OUTER, and FULL OUTER.

NOTE: *This step expects the rows coming in to be sorted on the specified key fields*

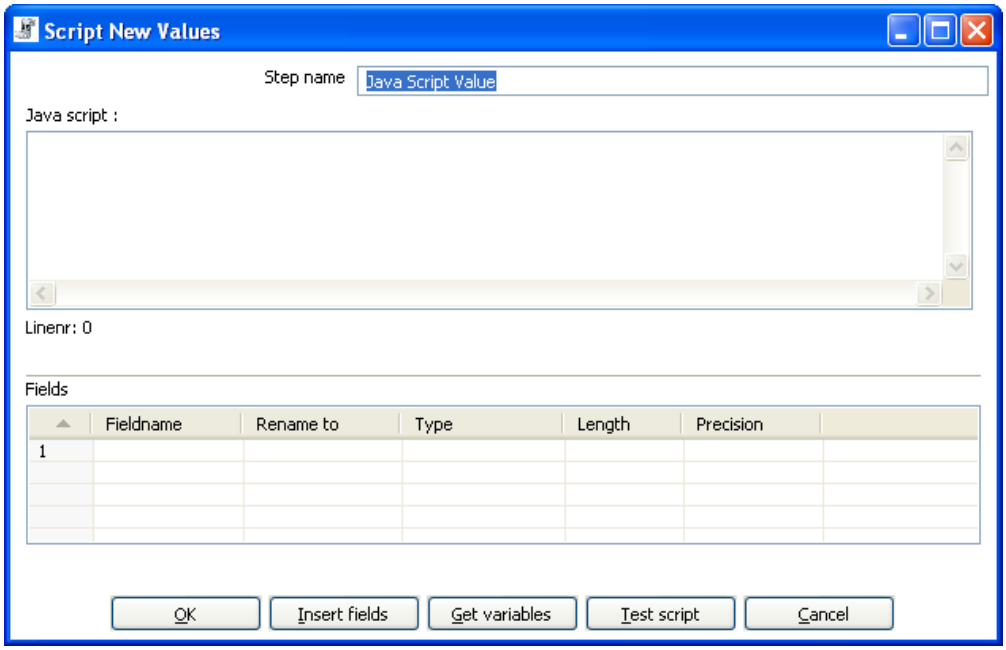
11.6.44.2. Options

The following table describes the options available for the Merge Join step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
First Step	Specify the first input step to the merge join.
Second Step	Specify the second input step to the merge join.
Join Type	Select from the available types of joins.
Keys for 1st step	Specify the key fields on which the incoming data is sorted. Click the 'Get key fields' button to retrieve a list of fields from the specified step.
Keys for 2nd step	Specify the key fields on which the incoming data is sorted. Click the 'Get key fields' button to retrieve a list of fields from the specified step.

11.6.45. JavaScript Values

Icon



Java Script value dialog

11.6.45.1. General description

This step type allows you to do complex calculations using the JavaScript language. The JavaScript engine used is Rhino 1.5R5.

11.6.45.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Java Script	Specify the script you wish to run.
Fields	These are the fields that we want to add to the output steam.
Insert fields button	Inserts the fields and the standard method to grab the value of the field.
Test script button	Tests whether or not the script compiles.
Get variables button	Gets the newly created variables and inserts them into the "Fields" grid.
Option	Description
Step name	Name of the step; this name has to be unique in a single transformation
Java Script	Specify the script you want to run
Fields	The fields to add to the output steam
Insert fields button	Inserts the fields and the standard method to grab the value of the field
Test script button	Tests whether or not the script compiles
Get variables button	Gets the newly created variables and inserts them into the Fields grid

11.6.45.3. Value functions

This is a list of functions that you can use to manipulate values:

Function	Description
Value Clone()	Builds a copy of a value and returns a Value.
void setName(String name)	Sets the name of a value.
String getName() .	Get the name of a value
void setValue(double num)	Set the value to a floating point value.
void setValue(String str)	Set the value to a string value.
void setValue(Date dat)	Set the value to a Date value.
void setValue(boolean bool)	Set the value to a Boolean value.
void setValue(long l)	Set the value to an integer value.
void setValue(Value v)	Set the value to the value contained in another field.
double getNumber()	Gets the value of a field as a floating point value.
String getString()	Gets the value of a field as a textual string representation.
int getStringLength()	Gets the length of the string representation.
Date getDate()	Gets the value of the field as a date value.
boolean getBoolean()	Gets the value of a field as a Boolean. **NOTE:** *String "Y" or "true" is converted to true.* **NOTE:** *Numeric value 0 is converted to false, everything else is true.*
long getInteger()	Gets the value of a field as an integer. **NOTE:** *Date fields are converted to the number of milliseconds since January 1st 1970 00:00:00 GMT.*
boolean isEmpty()	If the value has no type, this function returns true.
boolean isString()	If the value is of type String, this function returns true.
boolean isDate()	If the value is of type String, this function returns true.
boolean isNumber()	If the value is of type Number, this function returns true.
boolean isBoolean()	If the value is of type Boolean, this function returns true.
boolean isInteger()	If the value is of type Integer, this function returns true.
boolean isNumeric()	If the value is of type Number or Integer, this function returns true.
String toString()	Returns the textual representation of the value.
String toString(boolean pad)	Returns the textual representation of the value, padded to the length of the string if pad = true.
String toStringMeta()	Returns the meta-data information of the value as a string.
void setLength(int l)	Sets the length of the value.
void setLength(int l, int p)	Sets the length and precision of the value.
int getLength()	Returns the length of the value.
int getPrecision()	Returns the precision of the value.
void setPrecision(int p)	Sets the precision of a value.
String getTypeDesc()	Returns the description of a value as a string. (e.g. "String", "Number", "Date", "Integer", "Boolean").
void setNull()	Sets the value to Null.
void clearNull()	Removes the null setting.
void setNull(boolean n)	
boolean isNull()	Returns true if the value is null.
int compare(Value v)	Compares two values and returns 1 if the first value is larger than the second, -1 if it is smaller and 0 if they are equal.

Function	Description
boolean equals(Object v)	Compares two values and returns true if the two values have the same value.
int hashCode()	Returns a signed 64 values representing the value in the form of a hash code.
Value negate()	If the value is numeric, multiplies the value by -1, in all other cases it doesn't do anything.
Value and(Value v)	Calculates the bitwise AND of two integer values.
Value xor(Value v)	Calculates the bitwise XOR of two integer values.
Value or(Value v)	Calculates the bitwise OR of two integer values.
Value bool_and(Value v)	Calculates the boolean AND of two boolean values.
Value bool_or(Value v)	Calculates the boolean OR of two boolean values.
Value bool_xor(Value v)	Calculates the boolean XOR of two boolean values.
Value bool_not()	Calculates the boolean NOT of a boolean value.
Value greater_equal(Value v)	Compares two values and sets the first to true if the second is greater or equal to the first.
Value smaller_equal(Value v)	Compares two values and sets the first to true if the second is smaller or equal to the first.
Value different(Value v)	Compares two values and sets the first to true if the second is different from the first.
Value equal(Value v)	Compares two values and sets the first to true if the second is equal to the first.
Value like(Value v)	Sets the first value to true if the second string is part of the first.
Value greater(Value v)	Compares two values and sets the first to true if the second is greater than the first.
Value smaller(Value v)	Compares two values and sets the first to true if the second is smaller than the first.
Value minus(double v) Value minus(long v) Value minus(int v) Value minus(byte v) Value minus(Value v)	Subtracts v from the field value.
Value plus(double v) Value plus(long v) Value plus(int v) Value plus(byte v) Value plus(Value v)	Adds v to the field value.
Value divide(double v) Value divide(long v) Value divide(int v) Value divide(byte v) Value divide(Value v)	Divides the field value by v.
Value multiply(double v) Value multiply(long v) Value multiply(int v) Value multiply(byte v)	Multiplies the field value by v. **NOTE:** *Strings can be multiplied as well: the result is v times the string concatenated.*

Function	Description
Value multiply(Value v)	
Value abs()	Sets the field value to the –field value if the value was negative.
Value acos()	Sets the field value to the cosine of the number value.
Value asin()	Sets the field value to the arc sine of the number value.
Value atan()	Sets the field value to the arc tangents of the number value.
Value atan2(Value arg0) Value atan2(double arg0)	Sets the field value to the second arc tangents of the number value.
Value ceil()	Sets the field value to the ceiling of a number value.
Value cos()	Sets the field value to the cosine of a number value.
Value cosh()	Sets the field value to the hyperbolic cosine of a number value.
Value exp()	Sets the field value to the exp of a number value.
Value floor()	Sets the field value to the floor of a number value.
Value initcap()	Sets the all first characters of words in a string to uppercase. "matt casters" -> "Matt Casters"
Value length()	Sets the value of the field to the length of the String value.
Value log()	Sets the field value to the log of a number value.
Value lower()	Sets the field value to the string value in lowercase.
Value lpad(Value len) Value lpad(Value len, Value padstr) Value lpad(int len) Value lpad(int len, String padstr)	Sets the field value to the string value, left padded to a certain length. Default the padding string is a single space. Optionally, you can specify your own padding string.
Value ltrim()	Sets the field value to the string, without spaces to the left of the string.
Value mod(Value arg) Value mod(double arg0)	Sets the value to the modulus of the first and the second number.
Value nvl(Value alt)	If the field value is Null, set the value to alt.
Value power(Value arg)	
Value power(double arg0)	Raises the field value to the power arg.
Value replace(Value repl, Value with) Value replace(String repl, String with)	Replaces a string in the field value with another.
Value round()	Rounds the field value to the nearest integer.
Value rpad(Value len) Value rpad(Value len, Value padstr) Value rpad(int len) Value rpad(int len, String padstr)	Sets the field value to the string value, right padded to a certain length. Default the padding string is a single space. Optionally, you can specify your own padding string.
Value rtrim()	Remove the spaces to the right of the field value.
Value sign()	Sets the value of the string to -1, 0 or 1 in case the field value is negative, zero or positive.

Function	Description
Value sin()	Sets the value of the field to the sine of the number value.
Value sqrt()	Sets the value of the field to the square root of the number value.
Value substr(Value from, Value to) Value substr(Value from) Value substr(int from) Value substr(int from, int to)	Sets the value of the field to the substring of the string value.
Value sysdate() .	Sets the field value to the system date
Value tan(Value args[])	Sets the field value to the tangents of the number value.
Value num2str() Value num2str(String arg0) Value num2str(String arg0, String arg1) Value num2str(String arg0, String arg1, String arg2) Value num2str(String arg0, String arg1, String arg2, String arg3)	Converts a number to a string. Arg0: format pattern, see also Number Formats Arg1: Decimal separator (either . or ,) Arg2: Grouping separator (either . or ,) Arg3: Currency symbol For example converting value: `1234.56 using num2str("###,##0.00", ",", ".") gives 1.234,56` `.23 using num2str("###,##0.00", ",", ".") gives 0,23` `1234.56 using num2str("000,000.00", ",", ".") gives 001.234,56`
Value dat2str() Value dat2str(String arg0) Value dat2str(String arg0, String arg1)	Converts a date into a string. Arg0: format pattern, see also Number Formats Arg1: localized date-time pattern characters (u, t)
Value num2dat()	Converts a number to a date based upon the number of milliseconds since January 1st, 1970 00:00:00 GMT.
Value str2dat(String arg0) Value str2dat(String arg0, String arg1)	Converts a string to a date. Arg0: format pattern, see also Number Formats Arg1: localized date-time pattern characters (u, t)
Value str2num() Value str2num(String arg0) Value str2num(String arg0, String arg1) Value str2num(String arg0, String arg1, String arg2) Value str2num(String arg0, String arg1, String arg2, String arg3)	Converts a string into a number. Arg0: format pattern, see also Number Formats Arg1: Decimal separator (either . or ,) Arg2: Grouping separator (either . or ,) Arg3: Currency symbol
Value dat2num()	Converts a date into a number being the number of milliseconds since January 1st, 1970 00:00:00 GMT.
Value trim()	Remove spaces left and right of the string value.
Value upper()	Sets the field value to the uppercase string value.
Value e()	Sets the value to e
Value pi()	Sets the value to p

Function	Description
Value add_months(int months)	Adds a number of months to the date value.
Value last_day()	Sets the field value to the last day of the month of the date value.
Value first_day()	Sets the field value to the first day of the month of the date value.
Value trunc() Value trunc(double level) Value trunc(int level)	Set the field value to the truncated number or date value. Level means the number of positions behind the comma or in the case of a date, 5=months, 4=days, 3=hours, 2=minutes, 1=seconds, 0=miliseconds
Value hexEncode()	Encode a String value in its hexadecimal representation. E.g. If value is a string "a", the result would be "61".
Value hexDecode()	Decode a String value from its hexadecimal representation. E.g. If value is a string "61", the result would be "a". If the input string value is odd a leading 0 will be silently added.

11.6.45.4. JavaScript Examples

11.6.45.4.1. Remember the previous row

Sometimes it can be useful to know the value of the previous row. This can be accomplished by this piece of code:

var prev_row;

```
if (prev_row == null) prev_row = row;
...
String previousName = prev_row.getString("Name", "-");
...
prev_row = row;
```

Note that row is a special field that contains all values in the current row.

11.6.45.4.2. Set the location name of an address to uppercase

```
location.upper();
```

11.6.45.4.3. Extract information from a date field

// Year/Month/Day representation:

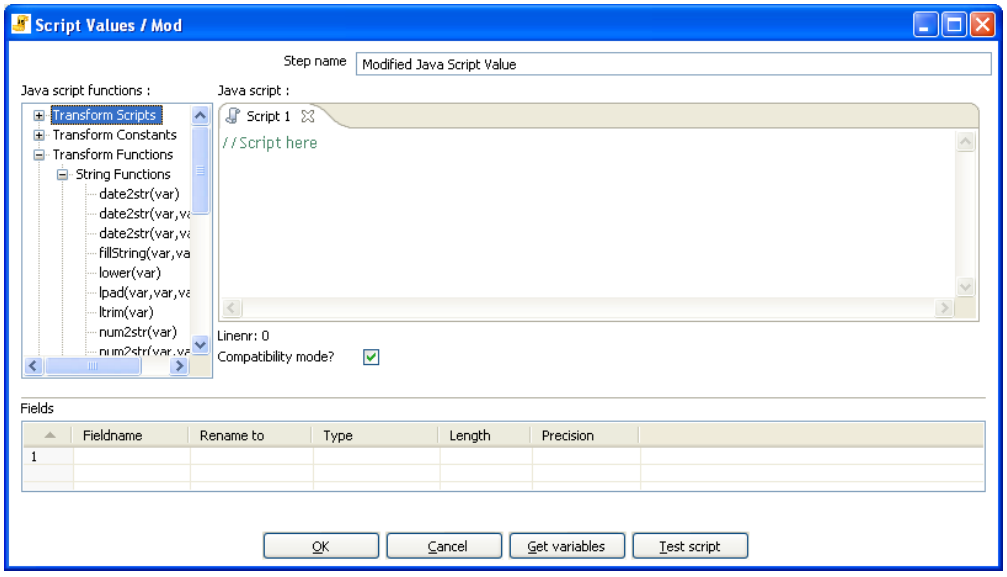
```
ymd = date_field.Clone().dat2str("yyyy/MM/dd").getString();
// Day/Month/Year representation:
dmy = date_field.Clone().dat2str("dd/MM/yyyy").getString();
// Year/Month :
ym = date_field.Clone().dat2str("yyyy/MM").getString();
// Long description of the month in the local language:
month_long_desc=
date_field.Clone().dat2str("MMMM").initcap().getString();
// Week of the year (1-53)
week_of_year = date_field.Clone().dat2str("w").getInteger();
// day of week, short description (MON-SUN)
day_of_week_short_desc
=date_field.Clone().dat2str("EEE").upper().getString();
```

```
// Day of the week (Monday-Sunday)
day_of_week_desc =
date_field.Clone().dat2str("EEEE").initcap().getString();
// Day of week (1-7)
day_of_week = date_field.Clone().dat2str("F").getInteger();
```

NOTE*: If you don't use Clone(), the original value will be overwritten by the methods that work on the Kettle Values.*

11.6.46. Modified Java Script Value

Icon



Modified Javascript values dialog

11.6.46.1. General Description

This is a modified version of the 'JavaScript Values' step that provides better performance and an easier, expression based user interface for building JavaScript expressions. This step will also allow you to create multiple scripts for each step.

11.6.46.2. Java script functions

This section provides a tree view of your available scripts, functions, input fields and output fields.

- **Transformation Scripts**: displays a list of scripts you have created in this step
- **Transformation Constants**: a list of pre-defined, static constants including SKIP_TRANSFORMATION, ERROR_TRANSFORMATION, and CONTINUE_TRANSFORMATION
- **Transformation Functions**: contains a variety of String, Numeric, Date, Logic and specialized functions you can use to create your script. To add a function to your script, simply double-click on the function or drag it to the location in your script that you wish to insert it.
- **Input Fields**: a list of inputs coming into the step. Double-click or use drag and drop to insert the field into your script.
- **Output Fields**: a list of outputs for the step.

11.6.46.3. Java Script

This section is where you edit the script for this step. You can insert functions, constants, input fields, etc. from the tree control on the left by double-clicking on the node you wish to insert or by dragging the object onto the Java Script panel.

11.6.46.4. Fields

The Fields table contains a list of variables from your script including the ability to add metadata like a descriptive name.

11.6.46.5. Extras

- **Get Variables button -** Retrieves a list of variables from your script.
- **Test script button -** Use this button to test the syntax of your script.

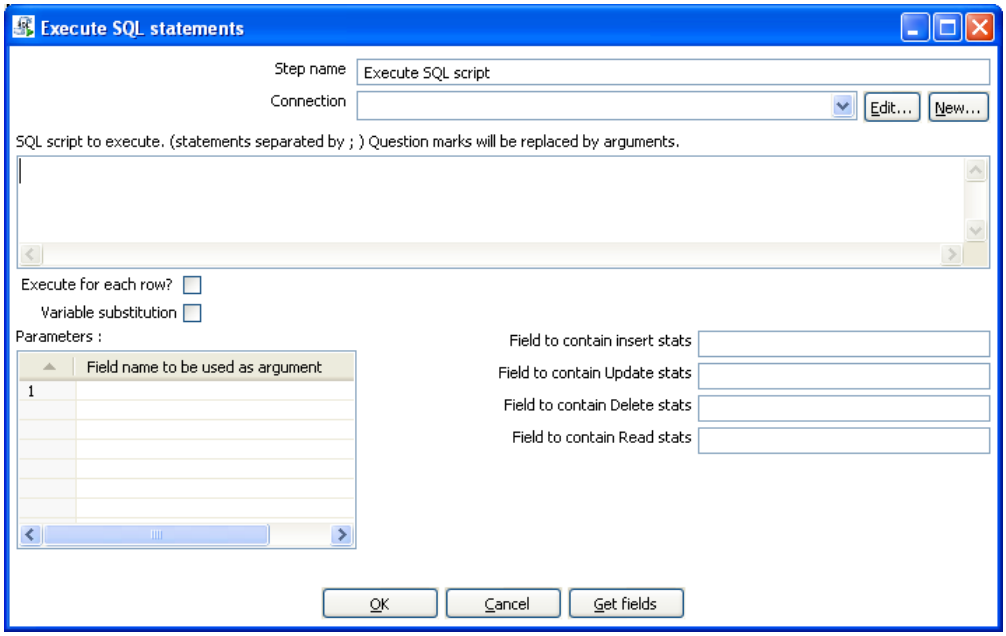
11.6.46.6. Java script internal API objects

You can use the following internal API objects (for reference see the classes in the source):

- **_TransformationName_**: a String with the actual transformation name
- **_step_**: the actual step instance of org.pentaho.di.trans.steps.scriptvalues_mod.ScriptValuesMod
- **rowMeta**: the actual instance of org.pentaho.di.core.row.RowMeta
- **row**: the actual instance of the actual data Object[]

11.6.47. Execute SQL script

Icon

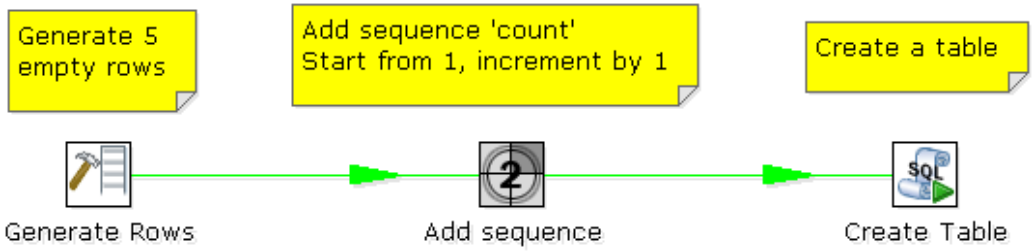


Execute SQL Scripts

11.6.47.1. General description

You can execute SQL scripts with this step, either once, during the initialization phase of the transformation, or once for every input-row that the step is given. The second option can be used to use parameters in SQL scripts.

For example, if you want to create 5 tables (tab1, tab2, tab3, tab4 and tab5) you could make such a transformation:



The SQL script to execute might look like this:

```
CREATE TABLE tab?
(
a INTEGER
);
```

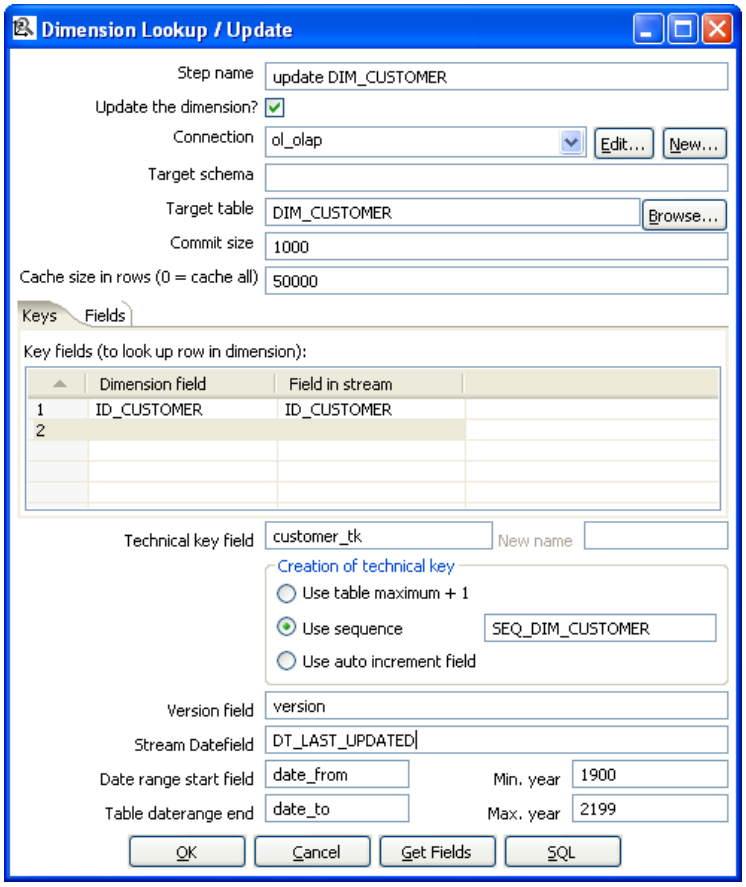
The field name to specify as parameter is then the "count" sequence we defined in the second step.

NOTE: *The execution of the transformation will halt when a statement in the script fails.*

As extra option, you can return the total number of inserts (INSERT INTO statements), updates (UPDATE table), deletes (DELETE FROM table) and reads (SELECT statements) by specifying the field names in the lower right of the dialog.

11.6.48. Dimension lookup/update

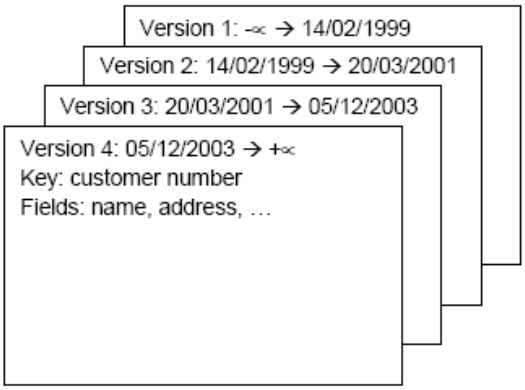
Icon



Dimension Lookup/Update

11.6.48.1. General description

This step type allows you to implement Ralph Kimball's slowly changing dimension for both types: Type I (update) and Type II (insert). Not only can you use this dimension for updating a dimension table, it can also be used for looking up values in a dimension.



Each dimension entry can be represented by a stack of papers containing the information valid during a certain period of time.
Insert then means that we add a new piece of paper containing the new information. (Type II)
Punch through means that we overwrite certain data on all pieces of paper for that certain customer number. (Type I)

In our dimension implementation each entry in the dimension table has the following properties:

Option	Description
Technical key	This is the primary key of the dimension.
Version field	Shows the version of the dimension entry (a revision number).
Start of date range	This is the fieldname containing the validity starting date.
End of date range	This is the fieldname containing the validity ending date.
Keys	These are the keys used in your source systems. For example: customer numbers, product id, etc.
Fields	These fields contain the actual information of a dimension.

As a result of the lookup or update operation of this step type, a field is added to the stream containing the technical key of the dimension. In case the field is not found, the value of the dimension entry for not found (0 or 1, based on the type of database) is returned.

NOTE: *This dimension entry is added automatically to the dimension table when the update is first run.*

11.6.48.2. Options

The following table provides a more detailed description of the options for the Dimension Lookup/Update step:

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Update the dimension?	Check this option if you want to update the dimension based on the information in the input stream. If this option is not enabled, the dimension only does lookups and only adds the technical key field to the streams.
Connection	Name of the database connection on which the dimension table resides.
Target schema	This allows you to specify a schema name to improve precision in the quoting and allow for table-names with dots '.' in it.
Target table	Name of the dimension table.
Commit size	Setting this to 10 will generate a commit every 10 inserts or updates.
Cache size in rows	This is the cache size in number of rows that will be held in memory to speed up lookups by reducing the number of round trips to the database. **Note:** *Please note that only the last version of a dimension entry is kept in memory. If there are more entries passing than what can be kept in memory, the technical keys with the highest values are kept in memory in the hope that these are the most relevant.* A cache size of 0 caches as many rows as possible and until your JVM runs out of memory. Use this option wisely with dimensions that can't grown too large.

Option	Description
	A cache size of -1 means that caching is disabled.
Keys tab	Specify the names of the keys in the stream and in the dimension table. This will enable the step to do the lookup.
Fields tab	For each of the fields you need to have in the dimension, you can specify whether you want the values to be updated (for all versions, this is a Type I operation) or you want to have the values inserted into the dimension as a new version. In the example we used in the screenshot the birth date is something that's not variable in time, so if the birth date changes, it means that it was wrong in previous versions. It's only logical then, that the previous values are corrected in all versions of the dimension entry.
Technical key field	This indicates the primary key of the dimension. It is also referred to as Surrogate Key. Use the new name option to rename the technical key after a lookup. For example, if you need to lookup different types of products like ORIGINAL_PRODUCT_TK, REPLACEMENT_PRODUCT_TK, ...
Creation of technical key	Specify howthe technical key is generated, options which are not available for your connection will be grayed out: • Use table maximum + 1: A new technical key will be created from the maximum key in the table. Note that the new maximum is always cached, so that the maximum does not need to be calculated for each new row. • Use sequence: Specify the sequence name if you want to use a database sequence on the table connection to generate the technical key (typical for Oracle e.g.). • Use auto increment field: Use an auto increment field in the database table to generate the technical key (typical for DB2 e.g.).
Version field	Specifies the name of the field to store the version (revision number) in.
Stream Datefield	If you have the date at which the dimension entry was last changed, you can specify the name of that field here. It allows the dimension entry to be accurately described for what the date range concerns. If you don't have such a date, the system date will be taken.
Date range start field	Specify the names of the dimension entries start range.
Table daterange end	Specify the names of the dimension entries end range.
Get Fields button	Fills in all the available fields on the input stream, except for the keys you specified.
SQL button	Generates the SQL to build the dimension and allows you to execute this SQL.

11.6.48.3. Remarks

- **For the Stream date field**: Consider adding an extra date field from System Info if you don't want the date ranges to be different all the time. For example if you have extracts from a source

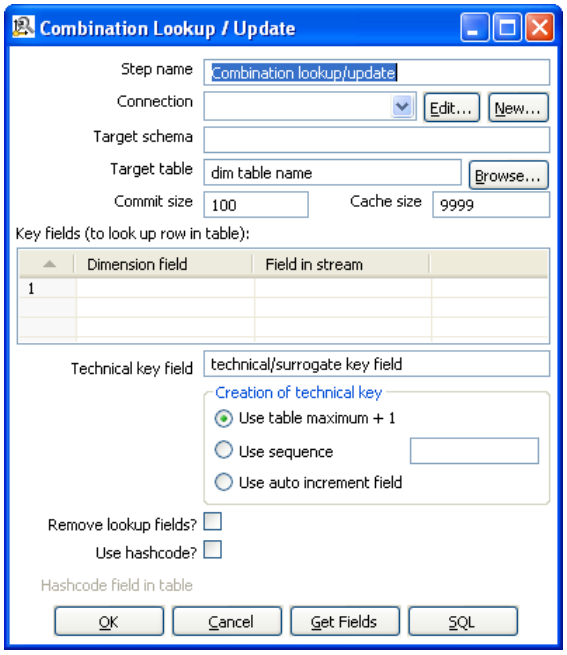
system being done every night at midnight, consider adding date "Yesterday 23:59:59" as a field to the stream by using a Join step.

IMPORTANT NOTE: *this needs to be a Date field. We isolate functionality and as such require you to do date type conversions in advance.*

- **For the "Date range start and end fields"**: You can only enter a year in these fields, not a timestamp. If you enter a year YYYY (e.g. 2100), it will be used as timestamp "YYYY-01-01 00:00:00.00" in the dimension table.

11.6.49. Combination lookup/update

Icon



Combination Lookup/Update

11.6.49.1. General description

This step allows you to store information in a junk-dimension table, and can possibly also be used to maintain Kimball pure Type 1 dimensions.

In short what it will do is:

1. Lookup combination of business key field1... fieldn from the input stream in a dimension table;
2. If this combination of business key fields exists, return its technical key (surrogate id);
3. If this combination of business key doesn't exist yet, insert a row with the new key fields and return its (new) technical key;
4. Put all input fields on the output stream including the returned technical key, but remove all business key fields if "remove lookup fields" is true.

So what this step does is create/maintain a technical key out of data with business keys. After passing through this step all of the remaining data changes for the dimension table can be made as updates, as either a row for the business key already existed or was created. This step will only maintain the key information, you will still need to update the non-key information in the dimension table, e.g. by putting an update step (based on technical key) after the combination update/lookup step.

Kettle will store the information in a table where the primary key is the combination of the business key fields in the table. Because this process can be very slow in case you have a large number of fields, Kettle also supports a "hash code" field representing all fields in the dimension. This can speed up lookup performance dramatically while limiting the fields to index to 1.

11.6.49.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Update the dimension?	Check this option if you want to update the dimension based on the information in the input stream. If this option is not enabled, the dimension only does lookups and only adds the technical key field to the streams.
Connection	Name of the database connection on which the dimension table resides.
Target schema	This allows you to specify a schema name to improve precision in the quoting and allow for table-names with dots '.' in it.
Target table	Name of the dimension table.
Commit size	Setting this to 10 will generate a commit every 10 inserts or updates.
Cache size in rows	This is the cache size in number of rows that will be held in memory to speed up lookups by reducing the number of round trips to the database. **Note:** *Please note that only the last version of a dimension entry is kept in memory. If there are more entries passing than what can be kept in memory, the technical keys with the highest values are kept in memory in the hope that these are the most relevant.* A cache size of 0 caches as many rows as possible and until your JVM runs out of memory. Use this option wisely with dimensions that can't grown too large. A cache size of -1 means that caching is disabled.
Key fields	Specify the names of the keys in the stream and in the dimension table. This will enable the step to do the lookup.
Technical key field	This indicates the primary key of the dimension. It is also referred to as Surrogate Key. Use the new name option to rename the technical key after a lookup. For example, if you need to lookup different types of products like ORIGINAL_PRODUCT_TK, REPLACEMENT_PRODUCT_TK, ...
Creation of technical key	Specify howthe technical key is generated, options which are not available for your connection will be grayed out: • Use table maximum + 1: A new technical key will be created from the maximum key in the table. Note that the new maximum is always cached, so that the maximum does not need to be calculated for each new row. • Use sequence: Specify the sequence name if you want to use a database sequence on the table connection to generate the technical key (typical for Oracle e.g.). • Use auto increment field: Use an auto increment field in the database table to generate the technical key (typical for DB2

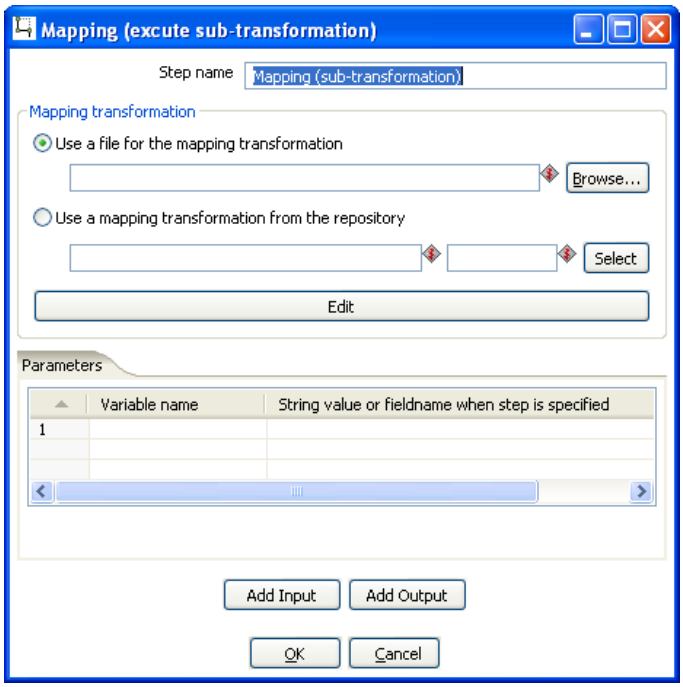
Option	Description
	e.g.).
Remove lookup fields?	Enable this option if you want to remove all the lookup fields from the input stream in the output. The only extra field added is then the technical key.
Use hashcode	This option allows you to generate a hash code, representing all values in the key fields in a numerical form (a signed 64 bit integer). This hash code has to be stored in the table. **IMPORTANT:** *This hash code is NOT unique. As such it makes no sense to place a unique index on it.*
Table daterange end	Specify the names of the dimension entries end range.
Get Fields button	Fills in all the available fields on the input stream, except for the keys you specified.
SQL button	Generates the SQL to build the dimension and allows you to execute this SQL.

11.6.49.3. Remarks

- The Combination Lookup/Update step assumes that the dimension table it maintains is not updated concurrently by other transformations/applications. When you use e.g. the "Table Max + 1" method to create the technical keys the step will not always go to the database to retrieve the next highest technical key. The technical will be cached locally, so if multiple transformations would update the dimension table simultaneously you will most likely get errors on duplicate technical keys. Using a sequence or an auto increment technical key to generate the technical key it is still not advised to concurrently do updates to a dimension table because of possible conflicts between transformations.
- It is assumed that the technical key is the primary key of the dimension table or at least has a unique index on it. It's not 100% required but if a technical key exists multiple times in the dimension table the result for the Combination Lookup/Update step is unreliable.

11.6.50. Mapping

Icon



Mapping (execute sub-transformation)

11.6.50.1. General description & use

When you need to do a certain transformation over and over again, you can turn the repetitive part into a mapping. A mapping is a transformation that:

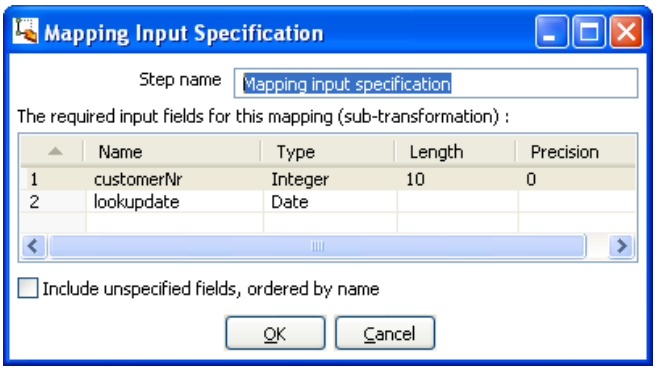
- specifies how the input is going to be using a MappingInput step
- specified how the input fields are transformed: the field that are added and deleted

For example you can create mappings for dimension lookups so that you don't need to enter the natural keys etc. every time again.

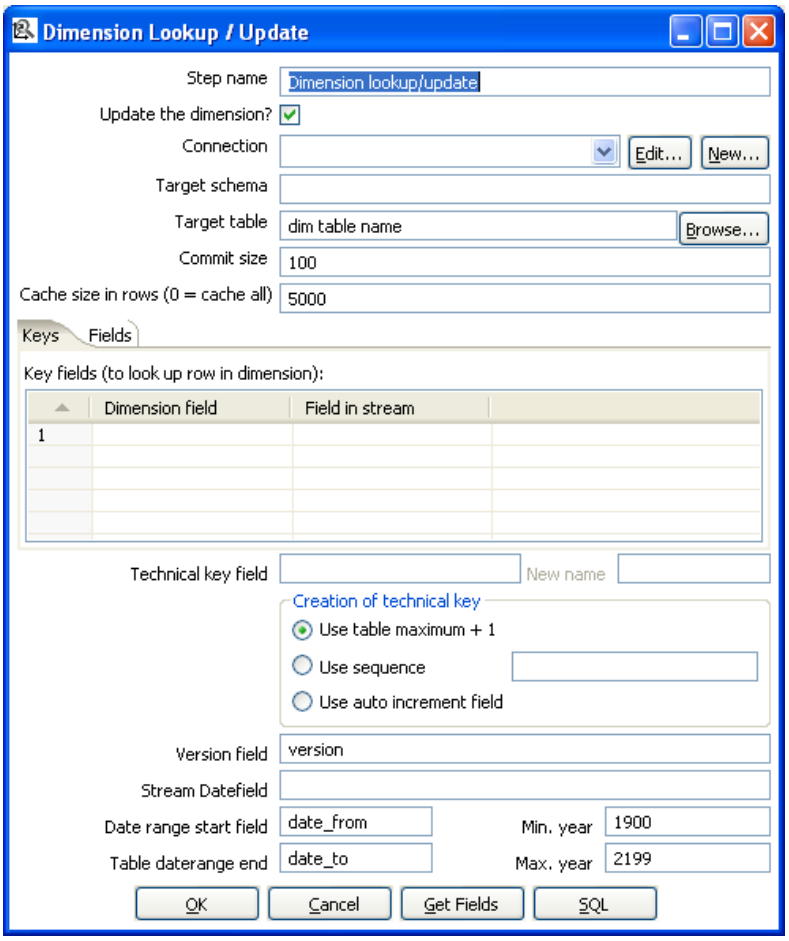
In the dialog, you can specify the transformation name and even launch another Spoon window to edit the selected transformation.

Example:

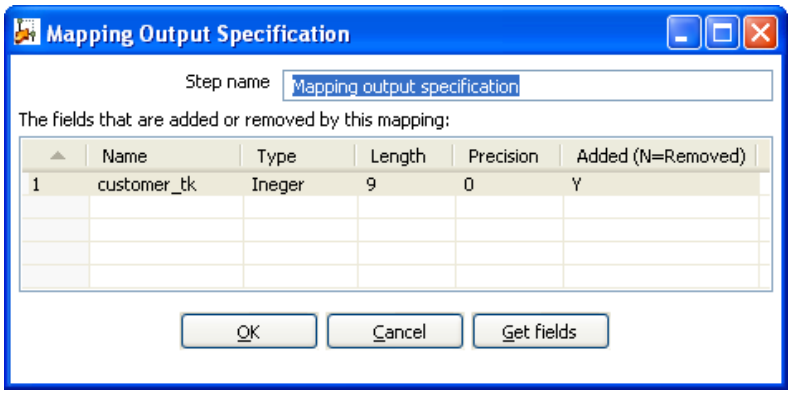
Suppose we want to create a mapping that does a lookup in the customer slowly changing dimension. In a larger warehouse, you need to specify the details for the dimension in question every time again. To get better re-use we want to create a mapping. The details needed for the dimension lookup are in this case the customer number and the lookup reference date. These 2 inputs we specify in the mapping input step:



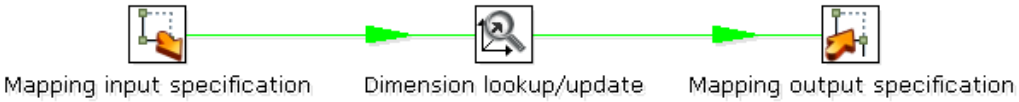
After this we can perform any calculation in our reusable transformation (Mapping), in our case we do a lookup in the dimension:



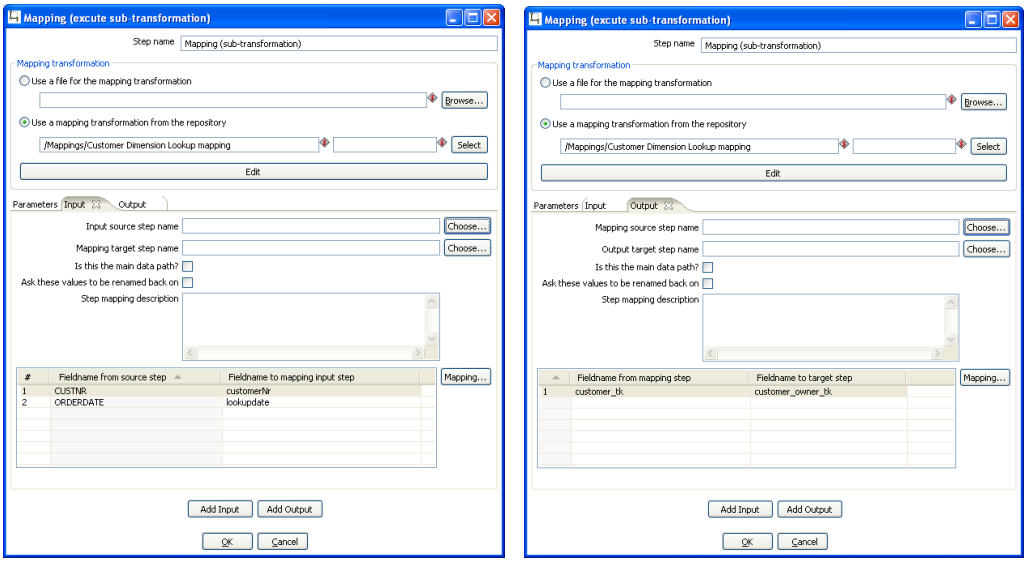
This dimension lookup step adds one field to the equation: customer_tk.
We can specify the fields that were in the Mapping Output step:



The complete mapping looks like this:



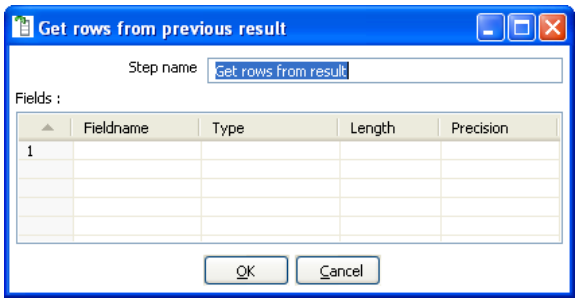
When we want to re-use this mapping, this is how we can do it:



As you can see, the way we do it is by "mapping" the stream fields of our choice to the required input fields of the mapping. Any added fields we can re-name on the output side.

11.6.51. Get rows from result

Icon



Get rows from previous result

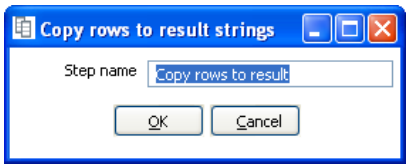
11.6.51.1. General description

This step returns rows that were previously generated by another transformation in a job.
The rows were passed on to this step using the "Copy rows to result" step.
To allow you to design more easily, you can enter the meta-data of the fields you're expecting from the previous transformation in a job.

IMPORTANT: *no validation of the supplied metadata is done at this time to allow for greater flexibility. It is just an aid at design time.*

11.6.52. Copy rows to result

Icon



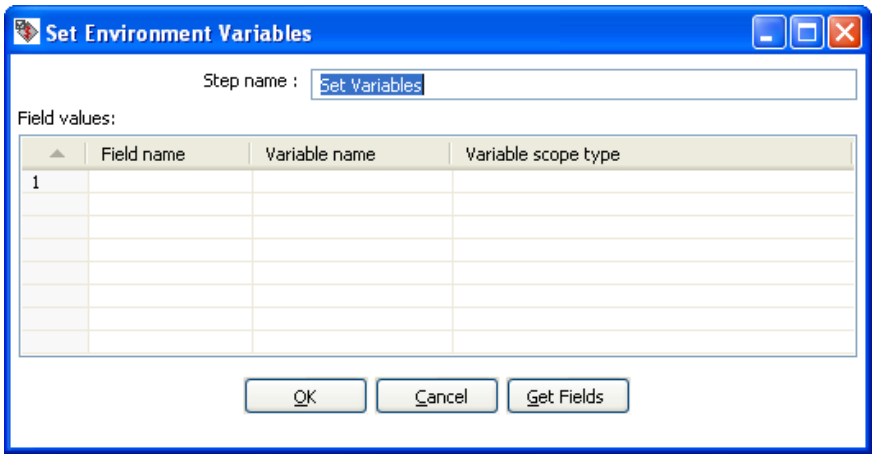
Copy rows to result strings

11.6.52.1. General description

This step allows you to transfer rows of data (in memory) to the next transformation (job entry) in a job.

11.6.53. Set Variable

Icon



Set Environment Variables

11.6.53.1. General description

This step allows you to set variables in a job or in the virtual machine. It accepts one (and only one) row of data to set the value of a variable.

Here are the possible scope settings:

- **Valid in the virtual machine**: the complete virtual machine will know about this variable.

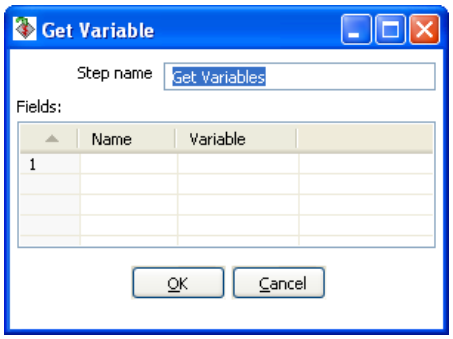
  WARNING: *this makes your transformation only fit to run in a stand-alone fashion. Running on an application server like on the Pentaho framework can become a problem. That is because other transformations running on the server will also see the changes this step makes.*
- **Valid in the parent job**: the variable is only valid in the parent job.
- **Valid in the grand-parent job**: the variable is valid in the grand-parent job and all the child jobs and transformations.
- **Valid in the root job**: the variable is valid in the root job and all the child jobs and transformations.

11.6.53.2. Variable usage

Refer to Variables for a description of the use of variables.

11.6.54. Get Variable

Icon



Get Variable

11.6.54.1. General description

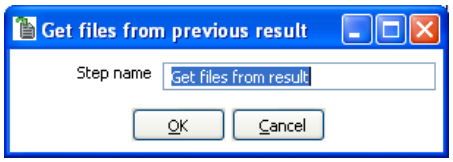
This step allows you to get the value of a variable. This step can return rows or add values to input rows.

NOTE: *You need to specify the complete variable specification in the format* `${variable}` *or* `%%variable%%` *(as described in Variables) . That means you can also enter complete strings in the variable column, not just a variable.*

For example, you can specify: `${java.io.tmpdir}/kettle/tempfile.txt` and it will be expanded to /tmp/kettle/tempfile.txt on Unix-like systems.

11.6.55. Get files from result

Icon



Get files from result

11.6.55.1. General description

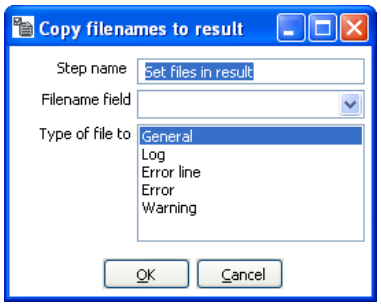
Every time a file gets processed, used or created in a transformation or a job, the details of the file, the job entry, the step, etc. is captured and added to the result. You can access this file information using this step.

These are the output fields:

Field name	Type	Example
type	String	Normal, Log, Error, Error-line, etc.
filename	String	somefile.txt
path	String	C:\Foo\Bar\somefile.txt
parentorigin	String	Process files transformation
origin	String	Text File Input
comment	String	Read by text file input
timestamp	Date	2006-06-23 12:34:56

11.6.56. Set files in result

Icon



Set file in result

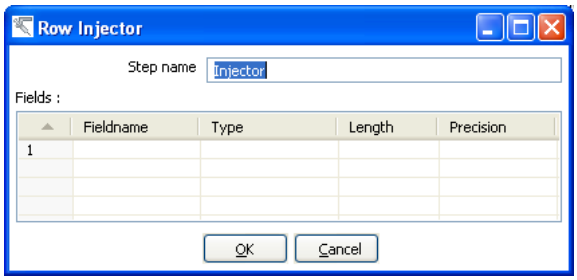
11.6.56.1. General description

This step can be used to route the list of files to the results stream.

For example, the Mail job entry can use this list of files to attach to a mail, so perhaps you don't want all files sent, but only a certain selection. For this, you can create a transformation that sets exactly those files you want to attach.

11.6.57. Injector

Icon



Row Injector

11.6.57.1. General description

- Injector was created for those people that are developing special purpose transformations and want to 'inject' rows into the transformation using the Kettle API and Java. Among other things you can build 'headless' transformations with it: transformations that have no input at design time: do not read from file or database.
- Here is some information on how to do it:
  - You can ask a Trans object for a RowProducer object
  - Also see the use case test in package: `be.ibridge.kettle.test.rowproducer`
  - Use this type of code:

```
Trans trans = new Trans(... TransMeta ...);
trans.prepareExecution(args);
RowProcuder rp = trans.addRowProducer(String stepname, int stepCopy);
```

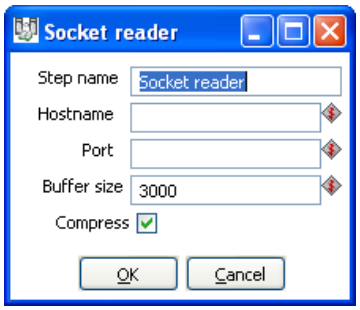
- After that you start the threads in the transformation. Then you can inject the rows while the transformation is running:

```
trans.startThreads();
...
rp.putRow(Row SomeRowYouHaveToInject);
...
```

- You can also specify the rows you are expecting to be injected. This makes it easier to build transformations because you have the meta-data at design time.

11.6.58. Socket reader

Icon



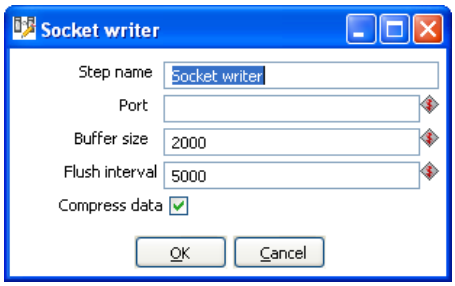
Socket reader

11.6.58.1. General description and use

Socket readers are used to transfer data from one server to another over TCP/IP. The primary use for these steps is in-line in a clustering environment. If you want to use these yourself, make sure to synchronize the preparation and start cycles of the transformations between the hosts. (like the clustered transformation does)

11.6.59. Socket writer

Icon



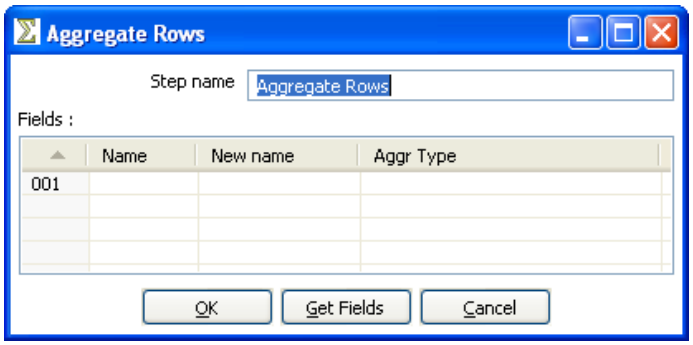
Socket writer

11.6.59.1. General description and use

Socket writers are used to transfer data from one server to another over TCP/IP. The primary use for these steps is in-line in a clustering environment. If you want to use these yourself, make sure to synchronize the preparation and start cycles of the transformations between the hosts. (like the clustered transformation does)

11.6.60. Aggregate Rows

Icon



Aggregate Rows

11.6.60.1. General description

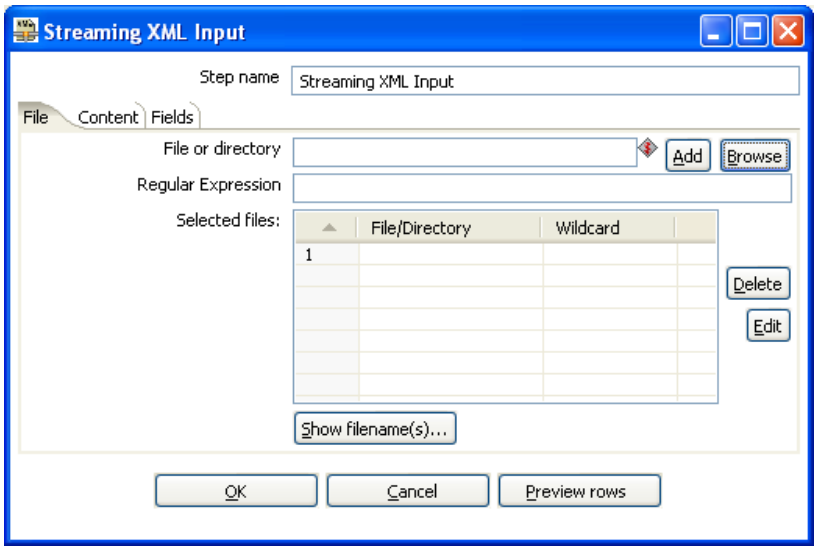
This step type allows you to quickly aggregate rows based upon all the rows. These are the available aggregation types:

- SUM - the sum of a field
- AVERAGE - the average of a field
- COUNT - the number of (not null) values
- MIN - the minimum value of a field
- MAX - the maximum value of a field
- FIRST - the first value of a field
- LAST - the last value of a field

NOTE: *This step type is deprecated. See the Group By step for a more powerful way of aggregating rows of data. The aggregate step can be removed in a future version.*

11.6.61. Streaming XML Input

Icon



Streaming XML

11.6.61.1. General description

The purpose of this step is to provide value parsing. This step is based on SAX parser to provide better performances with larger files. It is very similar to Xml Input, there are only differences in content and field tabs. The following sections describe in detail the properties and settings available for the Streaming XML input step.

11.6.61.2. File Tab

Option	Description
Step name	Name of the step. This name has to be unique in a single transformation.
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Show filenames(s)...	Displays a list of all files that will be loaded based on the current selected file definitions.

11.6.61.3. Content

Option	Description
Include filename in output & fieldname	Check this option if you want to have the name of the XML file to which the row belongs in the output stream. You can specify the name of the field where the filename will end up in.

Option	Description
Rownum in output & fieldname	Check this option if you want to have a row number (starts at 1) in the output stream. You can specify the name where the integer will end up in.
Limit	You can specify the maximum number of rows to read here.
Location	Specify the path by way of elements to the repeating part of the XML file. The element column is used to specify the element and position as follows: **A**: still specify an attribute **Ep**: specify an element defined by position (equivalent to E in original XMLInput). **Ea**: specify an element defined by an attribute and allow value parsing. Example: Ep=element/1 this is the first element called "element" Ea=element/att:val this is the element called "element" that have an attribute called "att" with "val" value

11.6.61.4. Fields

Option	Description
Name	Name of the field
Type	Type of the field can be either String, Date or Number
Format	See Number Formats for a complete description of format symbols.
Length	For *Number*: Total number of significant figures in a number; For *String:* total length of string; For *Date:* length of printed output of the string (e.g. 4 only gives back the year).
Precision	For *Number:* Number of floating point digits; For *String, Date, Boolean:* unused;
Currency	Used to interpret numbers like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10;000.00) or "," (5.000,00)
Group	A grouping can be a dot "," (10;000.00) or "." (5.000,00)
Trim type	type trim this field (left, right, both) before processing
Null if	treat this value as NULL
Repeat	Y/N: If the corresponding value in this row is empty: repeat the one from the last time it was not empty
Position	Position: The position of the XML element or attribute. You use the following syntax to specify the position of an element: The first element called "element": `E=element/1` The first attribute called "attribute": `A=attribute/1` The first attribute called "attribute" in the second "element" tag: `E=element/2, A=attribute/1` **NOTE**: *You can auto-generate all the possible positions in the XML file supplied by using the "Get Fields" button.* **NOTE**: *Support was added for XML documents where all the information is stored in the Repeating (or Root) element. The special R= locater was*

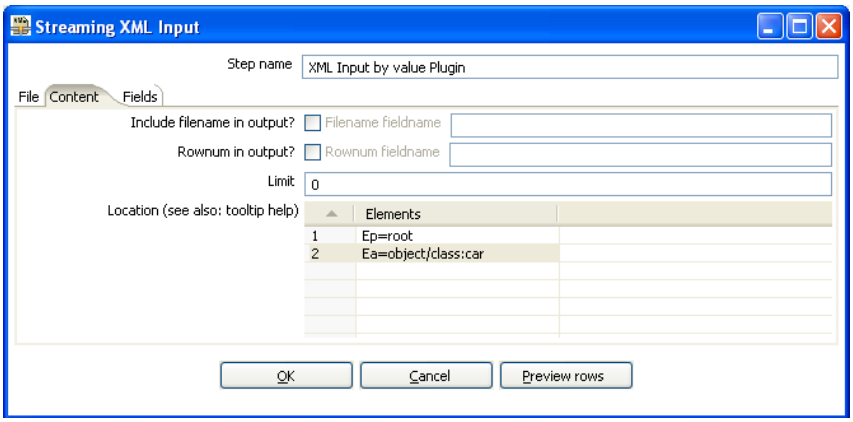
Option	Description
	added to allow you to grab this information. The "Get fields" button finds this information if it's present.

11.6.61.5. Streaming XML Example

Consider the following XML:

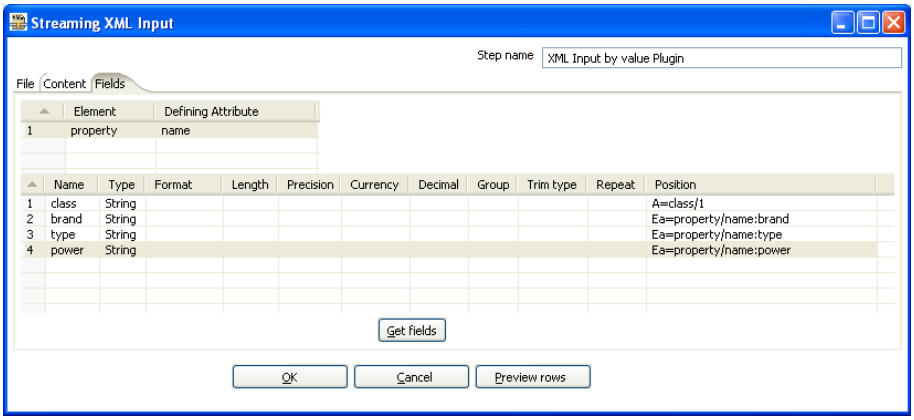
```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <root>
3    <object class="car">
4      <property name="brand">mercedes</property>
5      <property name="type">suv</property>
6      <property name="power">150hp</property>
7    </object>
8    <object class="plane">
9      <property name="brand">boeing</property>
10     <property name="type">747</property>
11     <property name="nr_places">400</property>
12   </object>
13   <object class="car">
14     <property name="brand">bmw</property>
15     <property name="type">truck</property>
16     <property name="power">248hp</property>
17   </object>
18 </root>
```

Suppose that we are interested in cars we must specify the location of the repeating element like this:



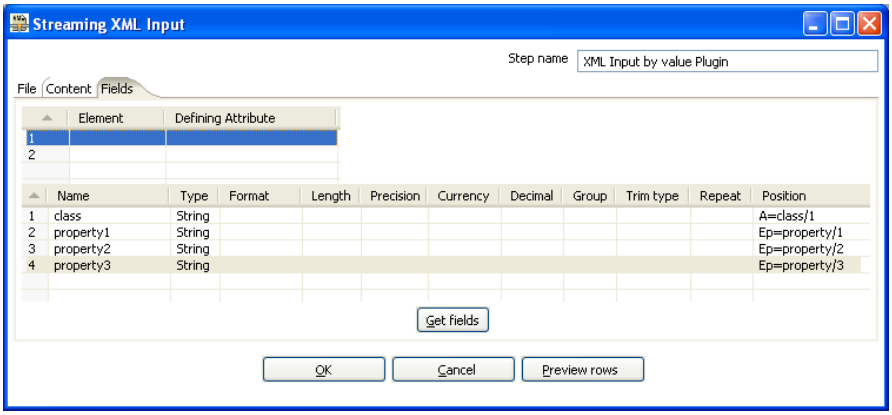
Now lets see the fields, we have different "property" elements that are differentiated by their "name" attribute, we are about to have the following fields "brand", "type" and "power" according to the "name" attribute.

For this, we must specify the association between "property" and "name" in the first grid.



Pressing the "Get Fields" button retrieves the right fields including properties.

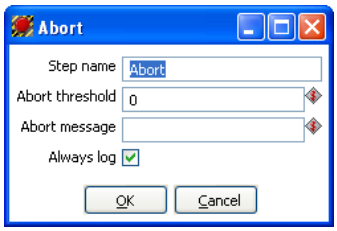
Let us now try leaving the new grid empty.



You can see that in this case the step is working like the original XMLInput and retrieve fields by their position. In this case, it is better to use value parsing, cause you get the right field names, and missing elements will not corrupt results (for example missing <property name="power"> </property> in some rows).

11.6.62. Abort

Icon



Abort

11.6.62.1. General description

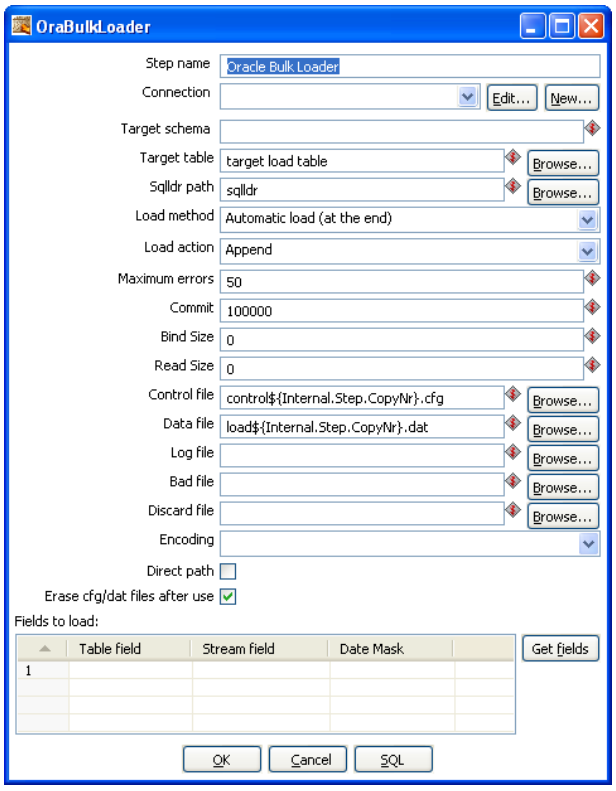
This step type allows you abort a transformation upon seeing input. It's main use is in error handling. For example, you can use this step so that a transformation can be aborted after x number of rows flow to over an error hop.

11.6.62.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Abort threshold	The threshold of number of rows after which to abort the transformations. E.g. If threshold is 0, the abort step will abort after seeing the first row. If threshold is 5, the abort step will abort after seeing the sixth row.
Abort message	The message to put in the log upon aborting. If not filled in a default message will be used.
Always log	Always log the rows processed by the Abort step. This allows the rows to be logged although the log level of the transformation would normally not do it. This way you can always see in the log which rows caused the transformation to abort.

11.6.63. Oracle Bulk Loader

Icon



Oracle Bulk Loader

11.6.63.1. General description

This step type allows you bulk load data to an Oracle database. It will write the data it receives to a proper load format and will then invoke Oracle SQL*Loader to transfer it to the specified table.

IMPORTANT! *Just like all steps in the "Experimental" category, this step is not considered ready for production use by the author. In the specific case of the Oracle Bulk loader we lacked the time to do extensive testing on it. Your feedback is most welcome as always.*

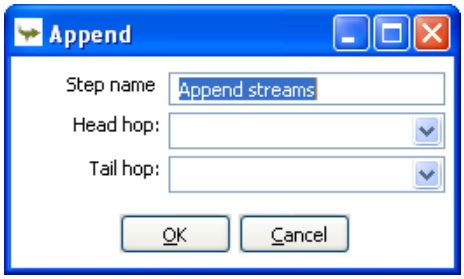
11.6.63.2. Options

Option	Description
Step name	Name of the step. ***Note:*** *This name has to be unique in a single transformation.*
Connection	Name of the database connection on which the dimension table resides.
Target schema	The name of the Schema for the table to write data to. This is important for data sources that allow for table names with dots '.' in it.
Target table	Name of the target table.
Sqldr path	Full path to the sqlldr utility (including sqlldr). If sqlldr is in the path of the executing application you can leave it to sqlldr.
Load method	Either "Automatic load (at the end)" or "Manual load (only creation of files)". Automatic load will start up sqlldr after receiving all input with the specified arguments in this step. Manual load will only create a control and data file, this

Option	Description
	can be used as a back-door: you can have PDI generate the data and create e.g. your own control file to load the data (outside of this step).
Load action	Append, Insert, Replace, Truncate. These map to the sqlldr action to be performed.
Maximum errors	The number of rows in error after which sqlldr will abort. This corresponds to the "ERROR" attribute of sqlldr.
Commit	The number of rows after which to commit, this corresponds to the "ROWS" attribute of sqlldr which differs between using a conventional and a direct path load.
Bind Size	Corresponds to the "BINDSIZE" attribute of sqlldr.
Read Size	Corresponds to the "READSIZE" attribute of sqlldr.
Control file	The name of the file used as control file for sqlldr.
Data file	The name of the data file in which the data will be written.
Log file	The name of the log file, optionally defined.
Bad file	The name of the bad file, optionally defined.
Discard file	The name of the discard file, optionally defined.
Encoding	Encodes data in a specific encoding, any valid encoding can be chosen besides the one in the drop down list.
Direct path	Switch on direct path loading, corresponds to DIRECT=TRUE in sqlldr.
Erase cfg/dat files after use	When switched on the control and data file will be erased after loading.
Fields to load	This table contains a list of fields to load data from, properties include: • **Table field**: Table field to be loaded in the Oracle table; • **Stream field**: Field to be taken from the incoming rows; • **Date mask**: Either "Date" or "Date mask", determines how date/timestamps will be loaded in Oracle. When left empty defaults to "Date" in case of dates.

11.6.64. Append

Icon



Append

11.6.64.1. General description

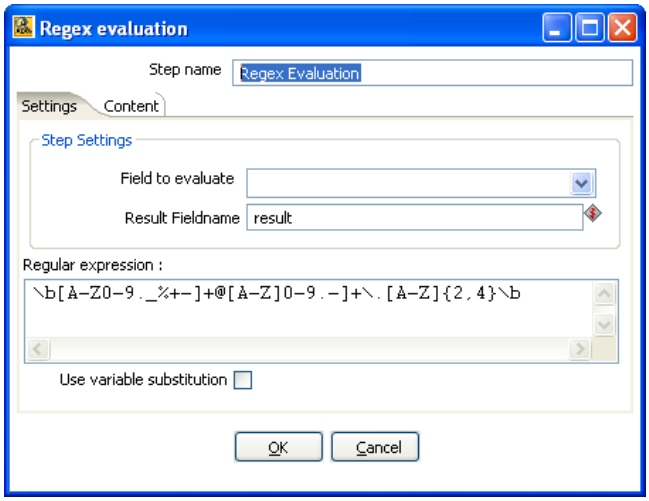
This step type allows you to order the rows of two inputs hops. First, all of the rows of the "Head hop" will be read and output, after that all of the rows of the "Tail hop" will be written to the output. If more than 2 hops need to be used you can use multiple append steps in sequence.

11.6.64.2. Options

Option	Description
Step name	Name of the step. ***Note:*** *This name has to be unique in a single transformation.*
Head hop	Name of the hop of which the rows should be output first.
Tail hop	Name of the hop of which the rows should be output last.

11.6.65. Regex Evaluation

Icon



Regex Evaluation

11.6.65.1. General description

This step type allows you to validate an input field against regular expression.
A regular expression (regex or regexp for short) is a special text string for describing a search pattern. For example, the equivalent regex for wildcard notations such as *.txt to find all text files in a file manager is:

```
.*\.txt
```

11.6.65.2. Settings Tab

Option	Description
Step name	Name of the step. ***Note:*** *This name has to be unique in a single transformation.*
Field to evaluate	Name of the field to evaluate
Result Fieldname	The name of the return field (boolean)
Regular expression	Put here the regular expression to match.
Use variable substitution	If you use variable, return it's content by selecting this option.

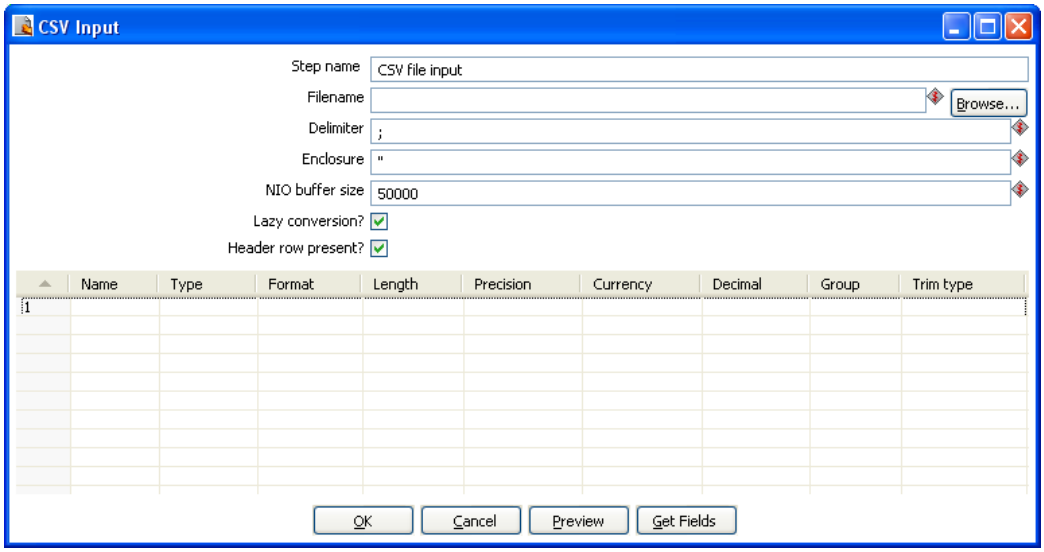
11.6.65.3. Content

Option	Description
Ignore differences in Unicode encodings	Check to ignore differences. **Note:** *This may improve performance, but be sure you data only contains US ASCII characters.*
Enables case-insensitive matching	By default, case-insensitive matching assumes that only characters in the US-ASCII charset are being matched. Unicode-aware case-insensitive matching can be enabled by specifying the 'Unicode-aware case...' flag in conjunction with this flag.

Option	Description
	Note: *You can also enable this via the embedded flag expression (?i).*
Permit whitespace and and comments in pattern	When enabled, the step will ignore whitespace and embedded comments starting with # through the end of the line. **Note:** *Comments mode can also be enabled via the embedded flag expression (?x).*
Enable dotall mode	When enabled, the expression '.' matches any character including the line terminator. By default, this expression does not match the line terminators. **Note:** *Dotall mode can also be enabled via the flag expression (?s).*
Enable multiline mode	When enabled, the expressions '^' and '$' match just after or just before, respectively, a line terminator or the end of the input sequence. By default, these expressions only match at the beginning and the end of the entire input sequence. **Note:** *Multiline mode can also be enabled via the flag expression (?m)*
Enable Unicode-aware case folding	When enabled, in conjunction with the Case-insensitive flag, case-insensitive matching is done in a manner consistent with the Unicode standard. By default, case-insensitive matching assumes that only characters in the US-ASCII charset are being matched. **Note:** *Unicode-aware case folding can also be enabled via the embedded flag expression (?u).*
Enables Unix lines mode	When enabled, only the line terminator is recognized in the behavior of '.', '^', and '$'. **Note:** *Unix lines mode can also be enabled via the embedded flag mode (?d).*

11.6.66. CSV Input

Icon



CSV Input

11.6.66.1. General description

This step provides the ability to read data from a delimited file. The following "standard" is used to read files: http://en.wikipedia.org/wiki/Comma-separated_values
However these are the exceptions to the rule:

1) Leading and trailing spaces or tabs, adjacent to commas, are NOT AUTOMATICALLY trimmed as this part of the standard seems to conflict with RFC-4180 as stated in the linked Wikipedia article. HOWEVER: we do allow trimming of fields to occur and this is in fact an automatically proposed option. As such I think we comply as much as possible.

2) We do not throw errors for missing "trailing" columns. We've had too many people file bugs against this behavior in the TFI step the last couple of years. Perhaps we can consider an option to throw errors if the format specified is not strictly present in the file. (Will be implemented in a next version upon change request)

3) We do not throw errors for escaping enclosures using a backspace. I don't think it's a good idea since we should still be able to read "\\\\". Either the backspace character has a special meaning or it doesn't. As far as I can tell in the CSV "standard" the backspace character has no special meaning.

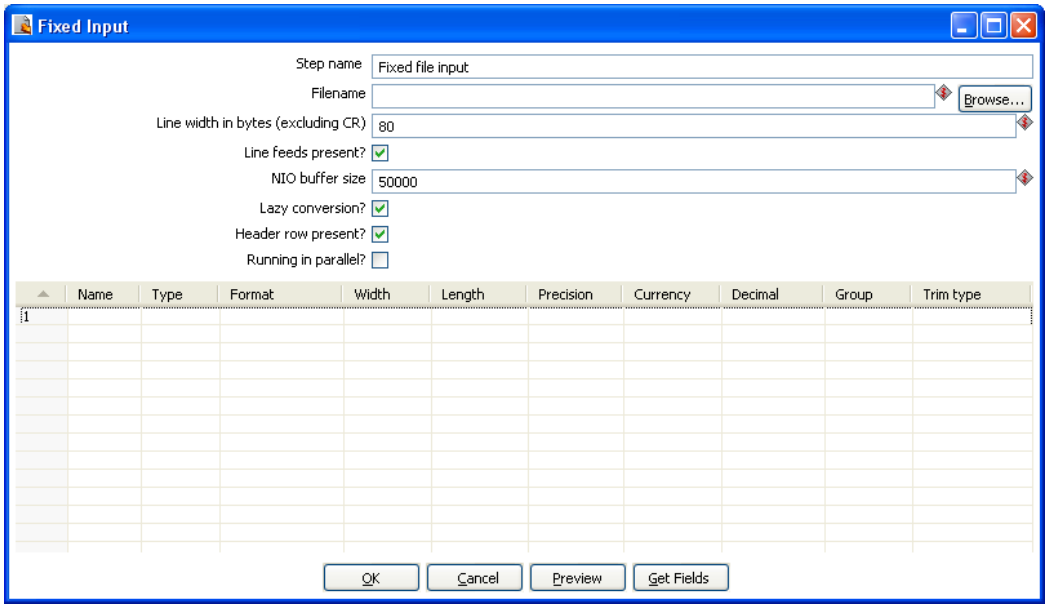
11.6.66.2. Options

The table below describes the options available for the CSV Input step:

Option	Description
Step name	Name of the step. ***Note:*** *This name has to be unique in a single transformation.*
Filename	Specify the name of the CSV file to read from.
Delimiter	Specify the file delimiter character used in the target file.
Enclosure	Specify the enclosure character used in the target file.
NIO buffer size	This step uses Non-Blocking I/O for increased performance. The buffer size is the number of bytes that will be read in one pass. Higher values typically lead to better performance, although it tops off quickly beyond a few MB. For *very* fast disks you might consider putting it in the >10MB range.
Lazy conversion	Lazy conversion will avoid unnecessary data type conversions and can result in a significant performance improvements. Check to enable.
Header row present?	Enable this option if the target file contains a header row containing column names.
Fields Table	This table contains an ordered list of fields to be read from the target file.
Preview button	Click to preview the data coming from the target file.
Get Fields button	Click to return a list of fields from the target file based on the current settings (i.e. Delimiter, Enclosure, etc.). All fields identified will be added to the Fields Table.

11.6.67. Fixed File Input

Icon



Fixed File Input

11.6.67.1. General description

This step is used to read data from a fixed width file

11.6.67.2. Options

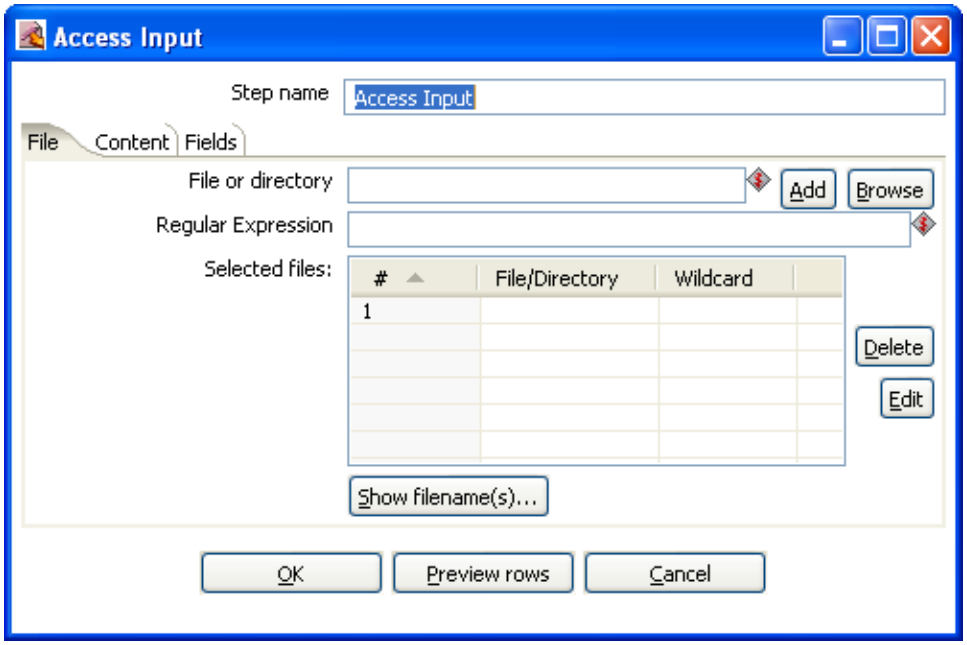
The table below describes the options available for the Fixed File Input step:

Option	Description
Step name	Name of the step. ***Note:*** *This name has to be unique in a single transformation.*
Filename	Specify the name of the CSV file to read from.
Line width in bytes	Specify the width of each record in the target file.
Line feeds present?	Check if the target file contains line feed characters.
NIO buffer size	This step uses Non-Blocking I/O for increased performance. The buffer size is the number of bytes that will be read in one pass. Higher values typically lead to better performance, although it tops off quickly beyond a few MB. For *very* fast disks you might consider putting it in the >10MB range.
Lazy conversion	Lazy conversion will avoid unnecessary data type conversions and can result in a significant performance improvements. Check to enable.
Header row present?	Enable this option if the target file contains a header row containing column names.
Running in parallel?	Enable this option to have each copy of the step read a dedicated part of the text file. For example, if you run the step in 5 copies locally, each step will read one 5th of the rows in the file. If you run it in 5 copies across 10 slave servers, each "Fixed Input" step will read one 50th of the rows in the file.
Fields Table	This table contains an ordered list of fields to be read from the target file.

Option	Description
Preview button	Click to preview the data coming from the target file.
Get Fields button	Click to return a list of fields from the target file based on the current settings (i.e. Delimiter, Enclosure, etc.). All fields identified will be added to the Fields Table.

11.6.68. Microsoft Access Input

Icon



Microsoft Access Input

11.6.68.1. General description

This step provides the ability to read data from a Microsoft Access database. The following sections describe the available options for the Access input step.

11.6.68.2. File Tab

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular Expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Show Filename(s) button	Displays a list of all files that will be loaded based on the current selected file definitions.
Preview rows button	Displays a preview of the data based on the current step configuration.

11.6.68.3. Content

Option	Description
Table	Specify the name of the table to read from or click browse to browse for a

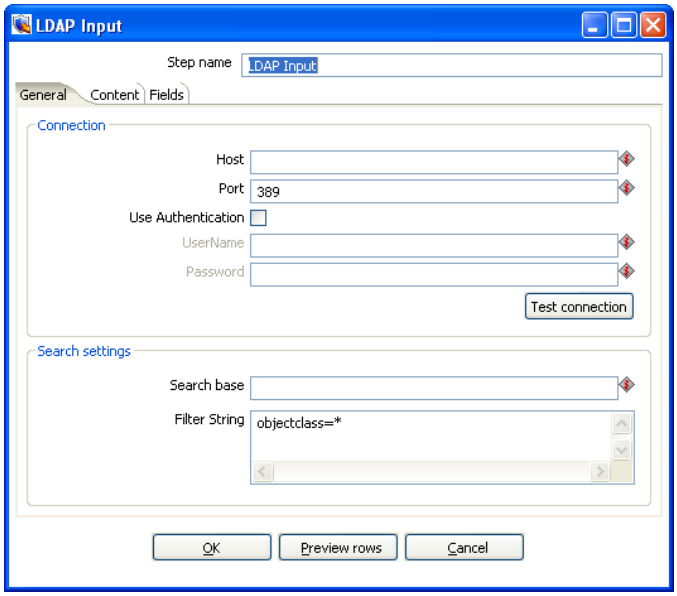
Option	Description
	table.
Include filename in output?	Optionally allows you to insert a field containing the filename onto the stream.
Include tablename in output?	Optionally allows you to insert a field containing the tablename onto the stream.
Include rownum in output?	Optionally allows you to insert a field containing the row number onto the stream.
Reset Rownum per file	Optionally allows you to reset the row number for each file being read from.
Limit	Optionally specify a limit on the number of rows to read.

11.6.68.4. Fields

Option	Description
Name	Name of the field
Column	The name of the column being read from.
Type	Type of the field can be either String, Date or Number
Format	See Number Formats for a complete description of format symbols.
Length	For *Number*: Total number of significant figures in a number; For *String:* total length of string; For *Date:* length of printed output of the string (e.g. 4 only gives back the year).
Precision	For *Number:* Number of floating point digits; For *String, Date, Boolean:* unused;
Currency	Used to interpret numbers like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10;000.00) or "," (5.000,00)
Group	A grouping can be a dot "," (10;000.00) or "." (5.000,00)
Trim type	Type trim this field (left, right, both) before processing
Repeat	Y/N: If the corresponding value in this row is empty: repeat the one from the last time it was not empty

11.6.69. LDAP Input

Icon



LDAP Input

11.6.69.1. General description

The LDAP Input step allows you to read information like users, roles and other data from an LDAP server. The following sections describe the available options for the LDAP input step.

11.6.69.2. General Tab

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Host	The hostname or IP address of the LDAP server.
Port	The TCP port to use. This is usually 389.
User Authentication	Enable this option if you want to pass authentication credentials to the LDAP server.
Username	Username for authenticating with the LDAP server.
Password	Password for authenticating with the LDAP server.
Test connection button	Click to test connecting to the LDAP server.
Search base	Specify the location in the directory from which the LDAP search begins.
Filter String	Specify the filter string for filtering the results.
Preview rows button	Click to preview rows based on the current step settings.

11.6.69.3. Content

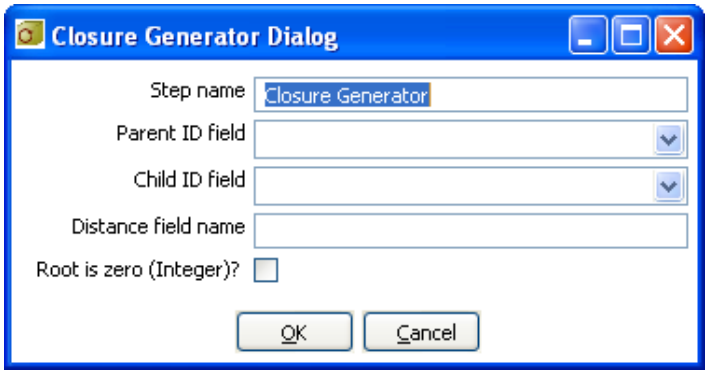
Option	Description
Include rownum in output?	Optionally allows you to insert a field containing the row number onto the stream.
Rownum fieldname	Specify the name of the field to contain row numbers.
Limit	Optionally specify the a limit on the number of rows to read.

11.6.69.4. Fields

Option	Description
Name	Name of the field
Column	The name of the column being read from.
Type	Type of the field can be either String, Date or Number
Format	See Number Formats for a complete description of format symbols.
Length	For *Number*: Total number of significant figures in a number; For *String:* total length of string; For *Date:* length of printed output of the string (e.g. 4 only gives back the year).
Precision	For *Number:* Number of floating point digits; For *String, Date, Boolean:* unused;
Currency	Used to interpret numbers like $10,000.00 or €5.000,00
Decimal	A decimal point can be a "." (10;000.00) or "," (5.000,00)
Group	A grouping can be a dot "," (10;000.00) or "." (5.000,00)
Trim type	Type trim this field (left, right, both) before processing
Repeat	Y/N: If the corresponding value in this row is empty: repeat the one from the last time it was not empty

11.6.70. Closure Generator

Icon



Closure Generator

11.6.70.1. General description

This step is used to generate a Reflexive Transitive Closure Table for Pentaho's Mondrian relational OLAP engine.

For more information on parent-child hierarchies in Mondrian and how closure tables can help improve performance, please refer to the Mondrian documentation found here.

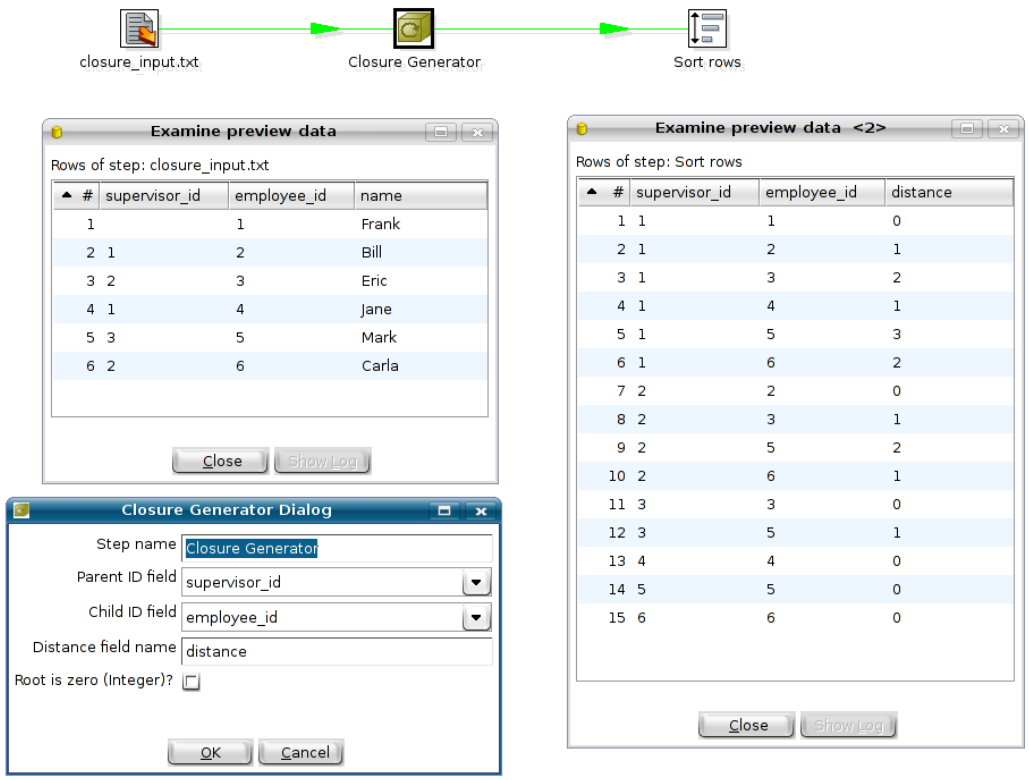
Technically, this step reads all input rows in memory and calculates all possible parent-child relationships. It attaches the distance (in levels) from parent to child.

11.6.70.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Parent ID field	The field name that contains the parent ID of the parent-child relationship.
Child ID field	The field name that contains the child ID of the parent-child relationship.
Distance field name	The name of the distance field that will be added to the output.
Root is zero (Integer)?	Check this box if the root of the parent-child tree is not empty (null) but zero (0).

11.6.70.3. Example

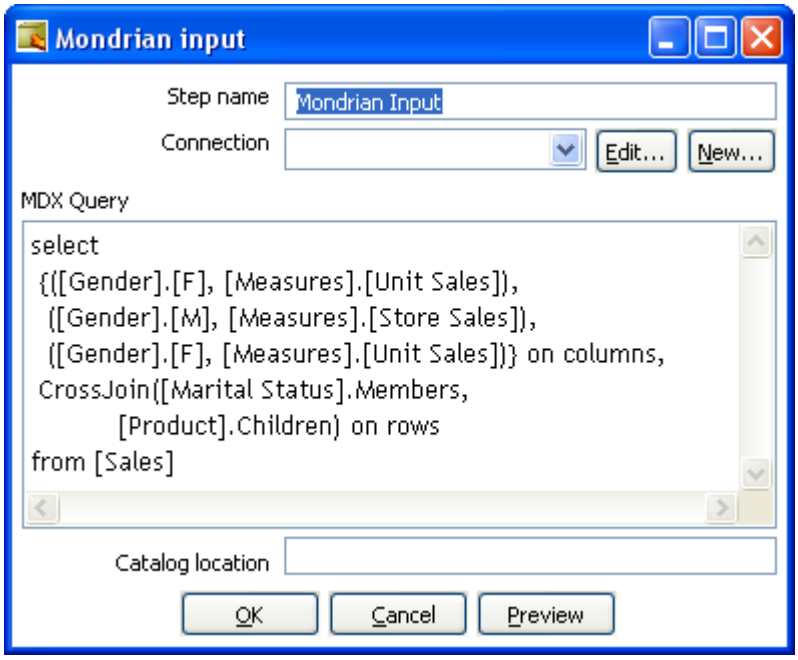
The example data shown below was taken from the Mondrian help pages on the subject of closure tables found here.



This transformation is available in directory samples/transformations/ in filename "Closure generator - standard mondrian sample.ktr".

11.6.71. Mondrian Input

Icon



Mondrian Input

11.6.71.1. General description

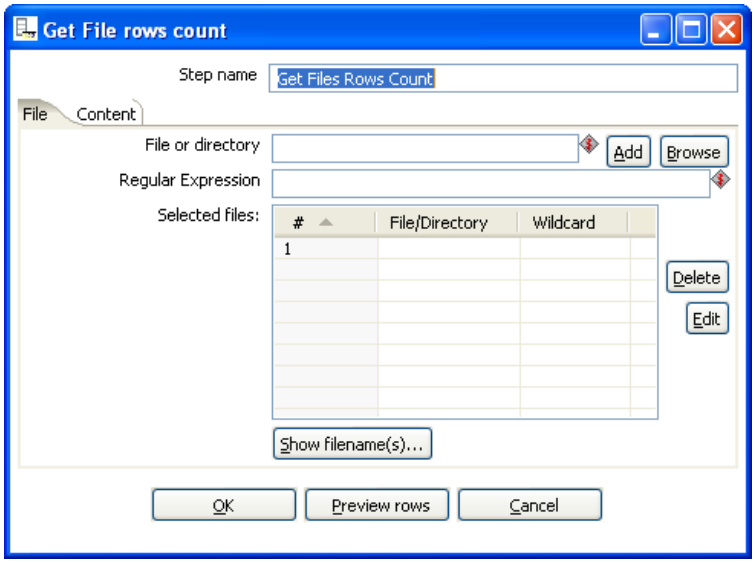
This step provides the ability to execute an MDX query against a Mondrian ROLAP cube and get the result back in a tabular format.

11.6.71.2. Options

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
Connection	The database connection to the database associated with the Mondrian cube you want to query.
MDX Query	Specify the MDX query you want to execute.
Catalog location	Specify the location of the Mondrian Schema file.
Preview button	Click to preview the data based on the current step settings.

11.6.72. Get Files Row Count

Icon



Get Files Row Count

11.6.72.1. General description

This step will return the row counts for one or more files. The following sections describe the available options for the Get Files Row Count step.

11.6.72.2. File Tab

Option	Description
Step name	Name of the step. **Note:** *This name has to be unique in a single transformation.*
File or directory	This field specifies the location and/or name of the input text file. **NOTE**: *press the "add" button to add the file/directory/wildcard combination to the list of selected files (grid) below.*
Regular Expression	Specify the regular expression you want to use to select the files in the directory specified in the previous option.
Selected Files	This table contains a list of selected files (or wildcard selections) along with a property specifying if file is required or not. If a file is required and it isn't found, an error is generated. Otherwise, the filename is simply skipped.
Show Filename(s)	Displays a list of all files that will be loaded based on the current selected file definitions.
Preview Rows	Displays the content of the selected file.

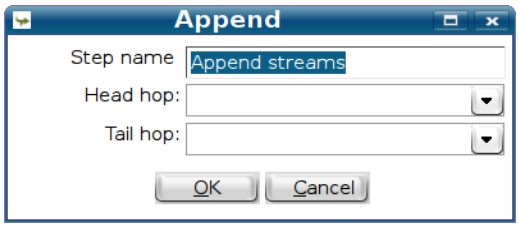
11.6.72.3. Content

Option	Description
Rows Count fieldname	Name of the field that will contain the file(s) row count(s).
Rows Separator type	Specify the row separator type for generating the row count.
Row separator	When the Separator type is set to custom, this setting is used to specify a custom row separator.
Include files count in	Optionally allows you to insert a field containing the file(s) count onto

output?	the stream.
Files Count fieldname	Name of the field that will contain the file counts.

11.6.73. Append streams

Icon



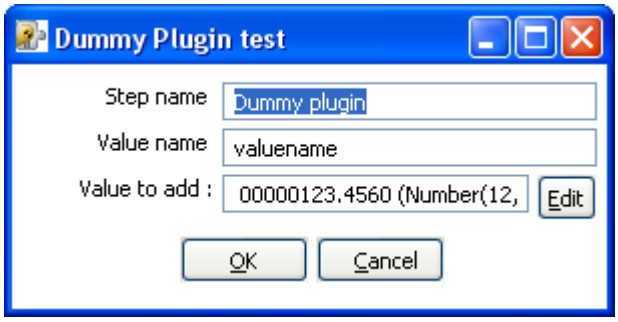
Append streams

11.6.73.1. General description

This step will first read and pass all the rows from the step specified as "Head hop" and then do the same for the "Tail hop". You can use this if you want to make sure that if you read from 2 separate files (Text File Input) the rows are read in a certain order.

11.6.74. Dummy Plugin

Icon

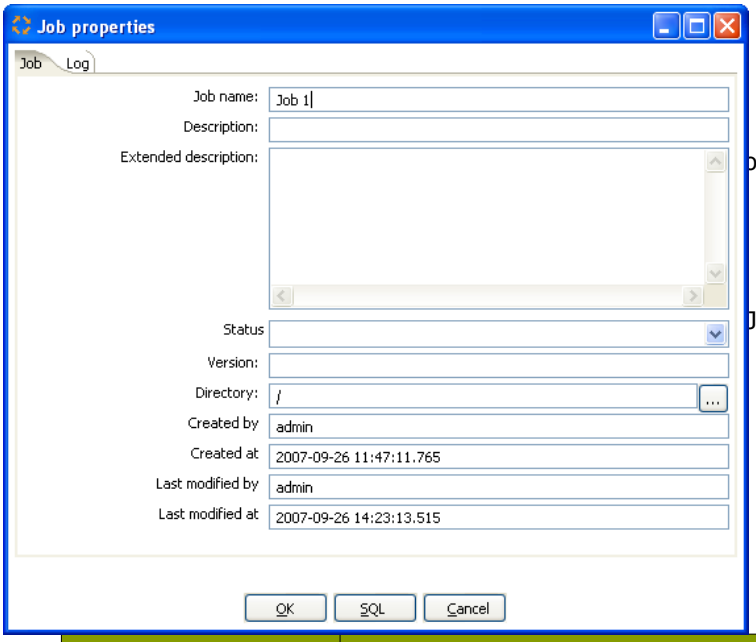


Dummy Plugin

11.6.74.1. General description

This step is provided as an example for developers on how to build a custom plug in. For more information on plugin development, refer to plug in development page found in the Pentaho documentation Wiki found here.

12. Job Settings



ow it is logging what it is doing. To access

Job tab:

Option	Description
Job Name	The name of the job. **Note:** *This is required information if you want to save to a repository.*
Description	Short description of the job, shown in the repository explorer
Extended description	Long extended description of the job
Status	Draft or production status
Version	Version description
Directory	The directory in the repository where the job is stored
Created by	Displays the original creator of the job.
Created at	Displays the date and time when the job was created.
Last modified by	Displays the user name of the last user that modified the job.
Last modified at	Displays the date and time when the job was last modified.

12.3. Log Tab

This table describes all of the general Job Settings found on the Log tab:

Option	Description
Log connection	Use this connection to write to a log table
Log table	specifies the name of the log table (for example L_ETL)
Use batch-ID	Use a batch ID in the logging table
Pass the batch-ID to job entries?	Check this if you want to pass the generated unique batch ID to (transformation) job entries in this job.
Use logfield to store logging in?	Check this if you want to store the logging of this job in the logging table in a long text field. (CLOB)
SQL button	Generates the SQL needed to create the logging table and allows you to execute this SQL statement.

13. Job Entries

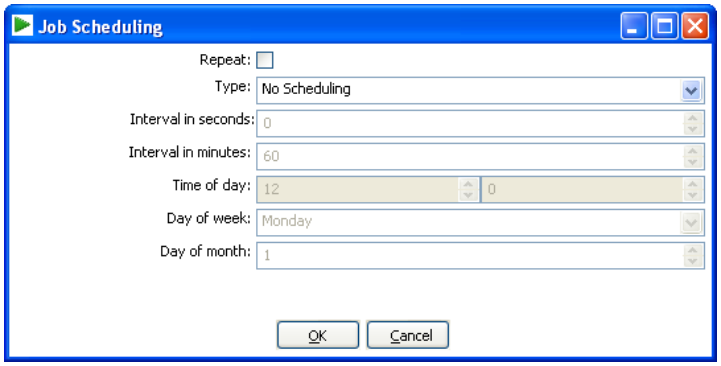
13.1. Description

A job entry is one part of a job. Job entries can provide you with a wide range of functionality ranging from executing transformations to getting files from a web server. Please see below for a complete list of all available job entry types.

13.2. Job Entry Types

13.2.1. Start

Icon



Start

13.2.1.1. General description

Start is where the job starts to execute and is required before the job can be executed. Only unconditional job hops are available from a Start job entry. The start icon also contains basic scheduling functionality.

13.2.2. Dummy Job Entry

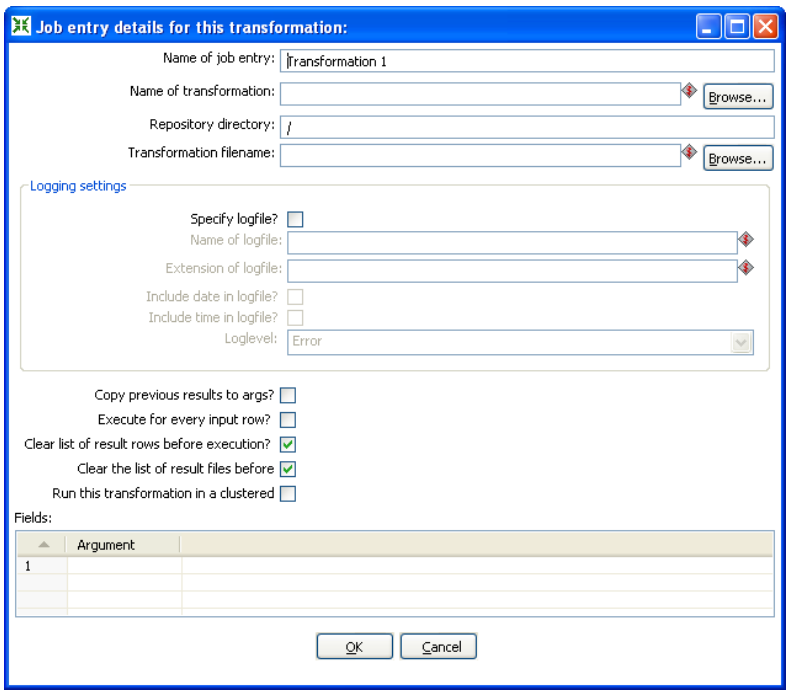
Icon

13.2.2.1. General Description

Use the Dummy job entry to do nothing in a job. This can be useful to make job drawings clearer or for looping. Dummy performs no evaluation.

13.2.3. Transformation

Icon



Start

13.2.3.1. General description

You can use the Transformation job entry to execute a previously defined transformation.

13.2.3.2. Options

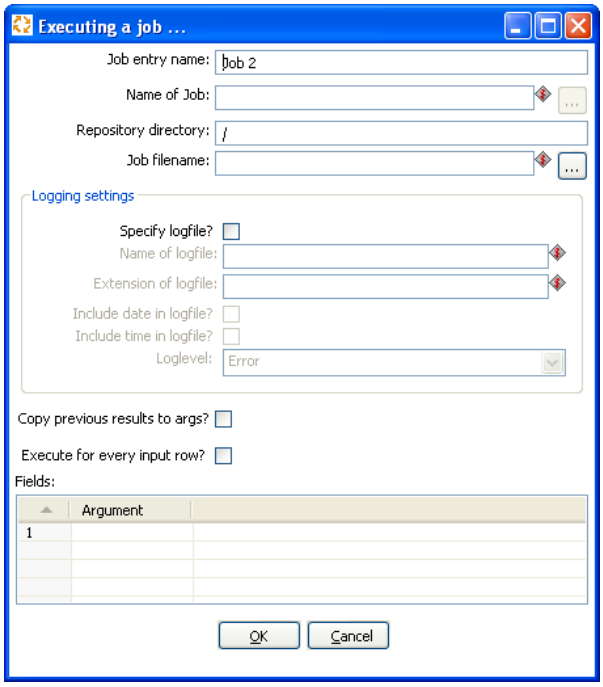
Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Name of transformation	The name of the transformation to start.
Repository directory	The directory in the repository where the transformation is located.
Filename	If you're not working with a repository, specify the XML filename of the transformation to start.
Specify log file	Check this if you want to specify a separate logging file for the execution of this transformation.
Name of log file	The directory and base name of the log file (for example C:\logs)
Extension of the log file	The filename extension (for example: log or txt)
Include date in filename	Adds the system date to the filename. (_20051231)
Include time in filename	Adds the system time to the filename. (_235959)
Logging level	Specifies the logging level for the execution of the transformation. See also the logging window in 15. Logging
Copy previous results to arguments	The results from a previous transformation can be sent to this one using the "Copy rows to result" step
Arguments	Specify the strings to use as arguments for the transformation.

Option	Description
Execute once for every input row	Support for "looping" has been added by allowing a transformation to be executed once for every input row.
Clear the list or result rows before execution	Checking this makes sure that the list or result rows is cleared before the transformation is started.
Clear the list of result files before execution	Checking this makes sure that the list or result files is cleared before the transformation is started.

NOTE: *you can use variables ${path} in the filename and transformation name fields to specify the transformation to be executed.*

13.2.4. Job

Icon



Job

13.2.4.1. General description

You can use the Job job entry to execute a previously defined job.

WARNING! *Although it is possible to create a recursive, never ending job that points to itself, you should be aware. This job will probably eventually fail with an out of memory or stack error.*

13.2.4.2. Options

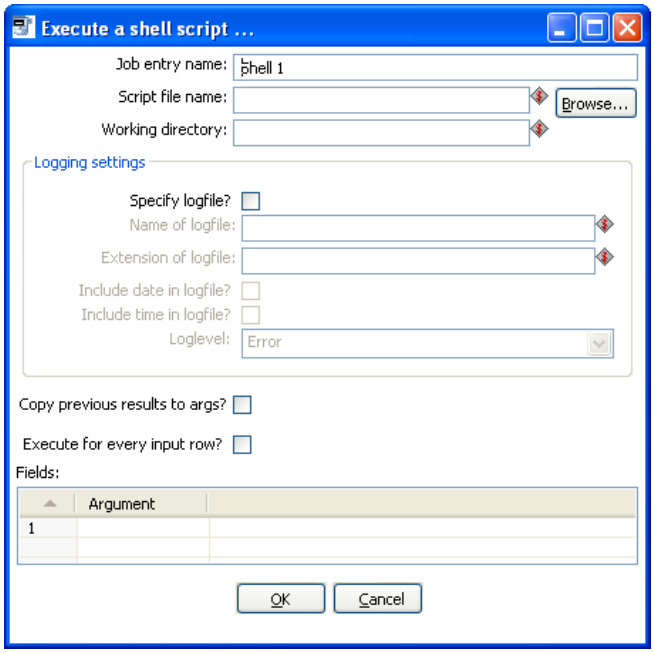
Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Name of transformation	The name of the job to start.
Repository directory	The directory in the repository where the job is located.
Filename	If you're not working with a repository, specify the XML filename of the job to start.
Specify log file	Check this if you want to specify a separate logging file for the execution of this job.
Name of log file	The directory and base name of the log file (for example C:\logs)
Extension of the log file	The filename extension (for example: log or txt)
Include date in filename	Adds the system date to the filename. (_20051231)

Option	Description
Include time in filename	Adds the system time to the filename. (_235959)
Logging level	Specifies the logging level for the execution of the job. See also the logging window in 15. Logging
Copy previous results to arguments	The results from a previous transformation can be sent to this job using the "Copy rows to result" step in a transformation.
Arguments	Specify the strings to use as arguments for the job.
Execute once for every input row	This implements looping. If the previous job entry returns a set of result rows, you can have this job executed once for every row found. One row is passed to this job at every execution. For example you can execute a job for each file found in a directory using this option.

NOTE: *you can use variables ${path} in the filename and job name fields to specify the job to be executed*.

13.2.5. Shell

Icon



Shell

13.2.5.1. General description

You can use the Shell job entry to execute a shell script on the host where the job is running.

NOTE: *Shell scripts can\output text to the console window. This output will be transferred to the Kettle logging system. Doing this no longer blocks the shell script.*
NOTE: *On Windows, scripts are preceded by "CMD.EXE /C" (NT/XP/2000) or "COMMAND.COM /C" (95,98).*

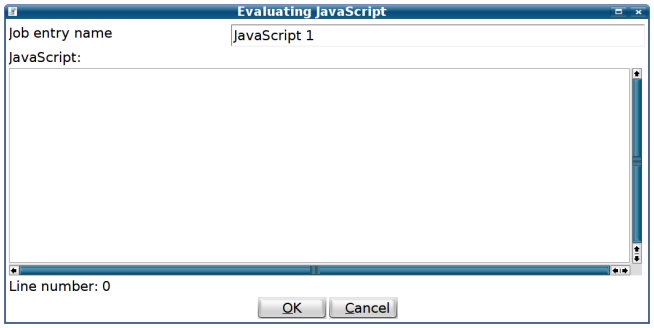
13.2.5.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Script file name	The filename of the shell script to start, should include full path else ${user.dir} is used as path.
Working directory	The directory that will be used as working directory for the shell script. The working directory only becomes active when the shell script starts so "Filename" should still include the full path to the script. When the field is left empty or the working directory is invalid ${user.dir} will be used as working directory.
Specify log file	Check this if you want to specify a separate logging file for the execution of this shell script.
Name of log file	The directory and base name of the log file (for example C:\logs)
Extension of the log file	The filename extension (for example: log or txt)

Option	Description
Include date in filename?	Adds the system date to the filename. (_20051231)
Include time in filename?	Adds the system time to the filename. (_235959)
Loglevel	Specifies the logging level for the execution of the shell. See also the logging window in 15. Logging
Copy previous results to arguments?	The results from a previous transformation can be sent to the shell script using the "Copy rows to result" step. (as arguments)
Execute once for every input row	This implements looping. If the previous job entry returns a set of result rows, you can have this shell script executed once for every row found. One row is passed to this script at every execution in combination with the copy previous result to arguments. The values of the corresponding result row can then be found on command line argument $1, $2, ... (%1, %2, %3, ... on Windows)
Arguments table	Specify the strings to use as arguments for the shell script.

13.2.6. JavaScript

Icon



JavaScript

13.2.6.1. General description

You can use the JavaScript entry (previously called Evaluation) to execute a JavaSript script to evaluate a boolean value. The result of the boolean value (true of false) will determine which next job entry will be evaluated if you are using condition hops.

13.2.6.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
JavaScript	The actual JavaScript to be evaluated. See also below for a more detailed description.

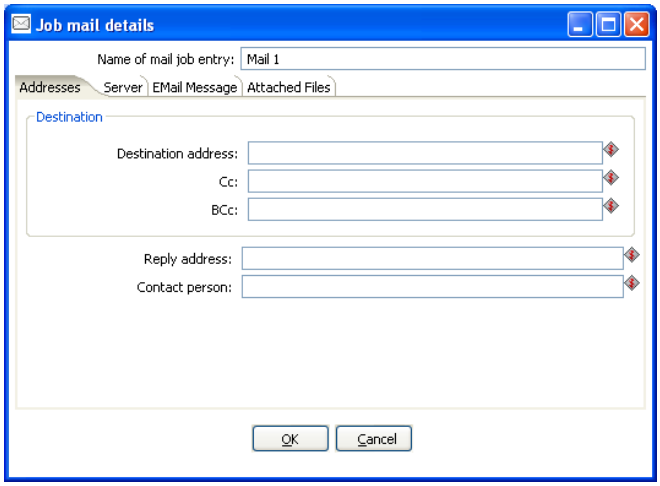
13.2.6.3. Exposed variables

These are the variables that are exposed to the JavaScript context. You can use these to evaluate the result of previous job entries to change the flow in your jobs.

Variable name	***Description / example***
errors	The number of errors that occurred in the previous job entry. Any value larger than 0 means that there was an error.
lines_input	The number of lines read from file, database or network.
lines_output	The number of lines written to file, database or network.
lines_updated	The number of lines updated in a file, a database or on the network.
lines_rejected	The number of lines rejected by error handling and routed to another step.
lines_read	The number of lines read from a previous step.
lines_written	The number of lines written to a next step.
files_retrieved	The number of files retrieved.
exit_status	The exit status of a previous shell job entry.
nr	This is a number that is incremented each time a job entry is executed. If you are doing loops you can use it to exit the loop after a number of repetitions.
is_windows	Boolean value that is true when you are running on any version of Microsoft Windows.
rows	The result rows set by a previous transformation. This is an array of Object type RowMetaAndData. You can use it to grab values from it like: ● var name = rows[0].getString("name", null); ● var size = rows[7].getInteger("size", -1);
parent_job	This is a handle to the parent job that is executing this JavaScript job entry. You can use it to do all kinds of interesting things, but the most interesting use case is getting and setting variables: ● var nrOfRows = parent_job.getVariable("NR_OF_ROWS"); ● parent_job.setVariable("FILENAME", "foo.txt");
previous_result	The result object of the previous job entry. It contains everything from the metrics and the result rows above to the result files list: ● var filesList = previous_result.getResultFilesList(); This is a list of all the files used or created in previous job entries.

13.2.7. Mail

Icon



Job Mail

13.2.7.1. General description

You can use the Mail job entry to send an e-Mail.

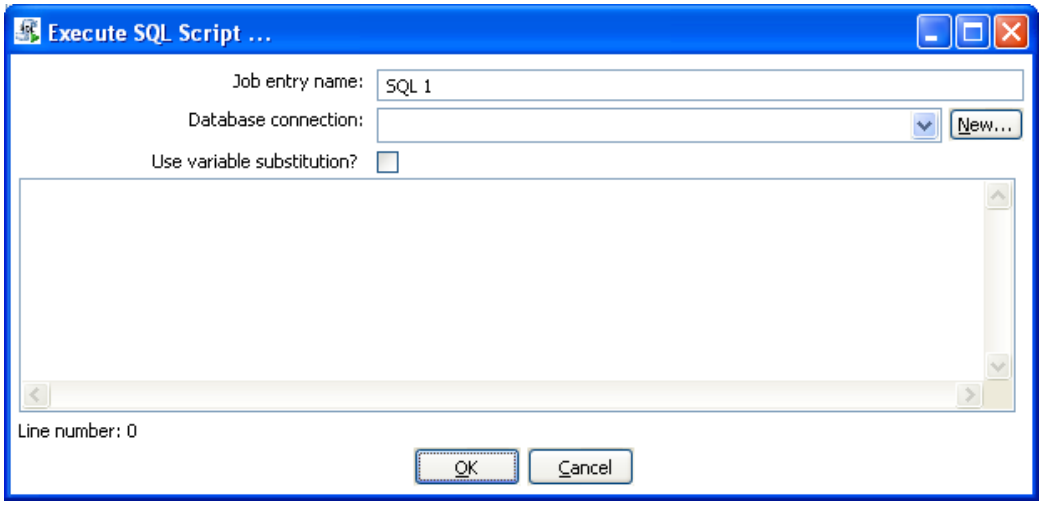
13.2.7.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Destination address	The destination for the e-Mail
Use authentication	Check this if your SMTP server requires you to authenticate yourself.
Authentication user	The user name to authenticate with
Authentication password	The password to authenticate with.
SMTP server	The mail server to which the mail has to be sent.
Reply address	The reply address for this e-Mail
Subject	The subject of the e-Mail
Include date in message	Check this if you want to include the date in the e-Mail
Contact person	The name of the contact person to be placed in the e-Mail
Contact phone	The contact telephone number to be placed in the e-Mail
Comment	Additional comment to be placed in the e-Mail
Attach files to message	Check this if you want to attach files to this message.
Select the result files types to attach.	When a transformation (or job) processes files (text, excel, dbf, etc) an entry is being added to the list of files in the result of that transformation or job. Specify the types of result files you want to add.
Zip files into a single archive	Check this if you want to zip all selected files into a single archive (recommended!) The zip filename Specify the name of the zip file that will be placed into the e-mail. **NOTE:** *All text fields can be specified using (environment and Kettle)*

Option	Description
	variables, possibly set in a previous transformation using the Set Variable step.

13.2.8. SQL

Icon



Execute SQL Script

13.2.8.1. General description

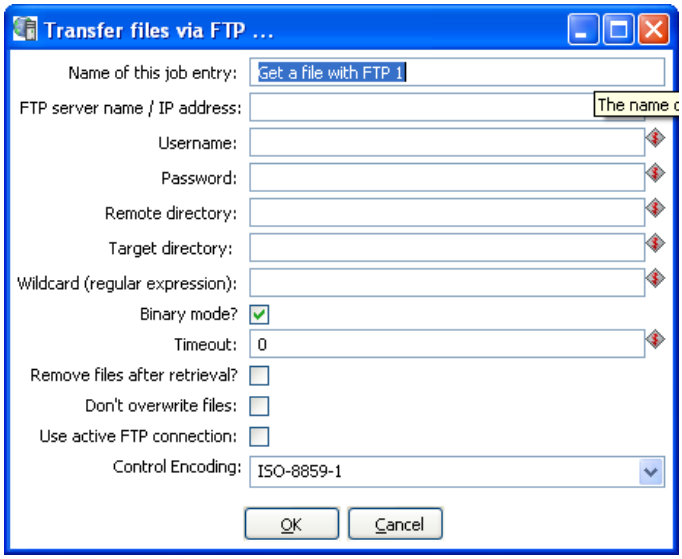
You can use the SQL job entry to execute an SQL script. This means a number of SQL statements separated by.

13.2.8.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Database Connection	The database connection to use.
Use variable substitution?	Enables kettle variables to be used in the SQL Script.
SQL script	The SQL script to execute.

13.2.9. Get a file with FTP

Icon



Get a file with FTP

13.2.9.1. General description

You can use the FTP job entry to get one or more files from an FTP server.

13.2.9.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
FTP server name	The name of the server or the IP address
User name	The user name to log into the FTP server
Password	The password to log into the FTP server
Remote directory	The remote directory on the FTP server from which we get the files
Target directory	The directory on the machine on which Kettle runs in which you want to place the transferred files
Wildcard	Specify a regular expression here if you want to select multiple files. For example: `.*txt$ : get all text files` `A.*[0-9]\.txt : files tarting with A ending with a number and .txt`
Use binary mode?	Check this if the files need to be transferred in binary mode.
Timeout	The FTP server timeout in seconds.
Remove files after retrieval?	Remove the files on the FTP server, but only after all selected files have been successfully transferred.
Don't overwrite files	Skip a file when a file with identical name already exists in the target directory.

Option	Description
Use active FTP connection	Check this to use active mode FTP instead of the passive mode (default).
Control Encoding	The encoding to use for the ftp control instructions, the encoding matters e.g. for the ftp'ing of filenames when they contain special characters. For Western Europe and the USA "ISO-8859-1" should suffice. You can enter any encoding that is valid on your server.

13.2.9.3. Notes

Some FTP servers do not allow files to be FTP'ed when they contain certain characters (spaces for example). Therefore, when choosing filenames for files to be FTP'ed, be sure to check up front whether your particular FTP server is able to process your kind of filenames.

13.2.10. Table Exists

Icon

Table Exists

13.2.10.1. General description

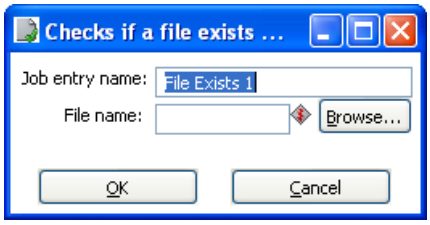
You can use the Table Exists job entry to verify if a certain table exists on a database.

13.2.10.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Database connection	The database connection to use
Table name	The name of the database table to check

13.2.11. File Exists

Icon



File Exists

13.2.11.1. General description

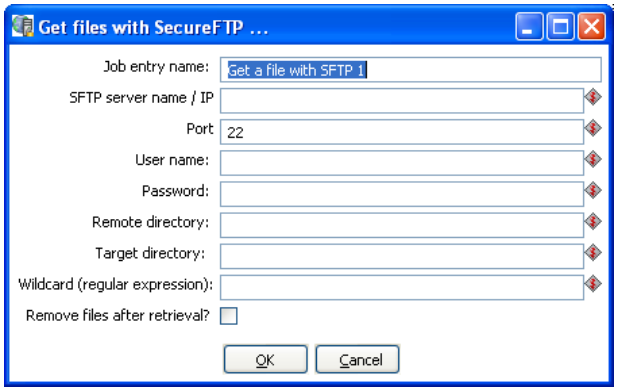
You can use the File Exists job entry to verify if a certain file exists on the server on which Kettle runs.

13.2.11.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Filename	The name and path of the file to check for

13.2.12. Get a file with SFTP

Icon



Get files with SecureFTP

13.2.12.1. General description

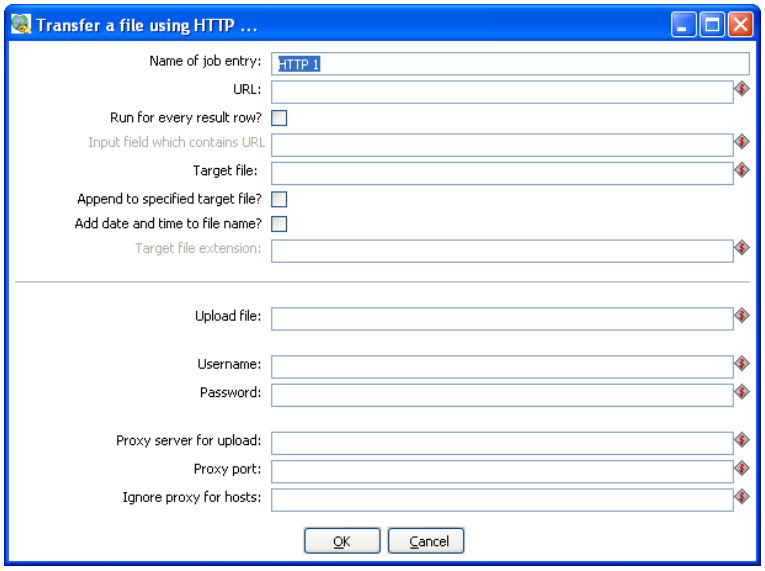
You can use the SFTP job entry to get one or more files from an FTP server using the Secure FTP protocol.

13.2.12.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
SFTP-server name / IP	The name of the SFTP server or the IP address
Port	The TCP port to use. This is usually 22
User name	The user name to log into the SFTP server
Password	The password to log into the SFTP server
Remote directory	The remote directory on the SFTP server from which we get the files
Target directory	The directory on the machine on which Kettle runs in which you want to place the transferred files
Wildcard	Specify a regular expression here if you want to select multiple files. For example: `.*txt$ : get all text files` `A.*[0-9]\.txt : files tarting with A ending with a number and .txt`
Remove files after retrieval?	Remove the files after they have been successfully transferred.

13.2.13. HTTP

Icon



HTTP Transfer

13.2.13.1. General description

You can use the HTTP job entry to get a file from a web server using the HTTP protocol.

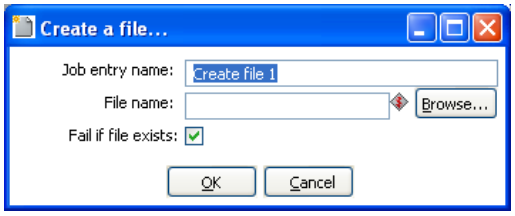
13.2.13.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
URL	The URL to use (for example: http://kettle.pentaho.org)
Run for every result row	Check this if you want to run this job entry for every row that was generated by a previous transformation. Use the "Copy rows to result"
Input field which contains URL	The fieldname in the result rows to get the URL from
Target filename	The target filename.
Append to specified target file	Append to the target file if it already exists
Add date and time to target filename	Check this if you want to add date and time yyyMMdd_HHmmss to the target filename.
Target filename extension	Specify the target filename extension in case you're adding a date and time to the filename
Upload file	
Username	The username to authenticate with. For Windows Domains, put the Domain in from of the user like this DOMAIN\Username
Password	The password to authenticate with.
Proxy server for upload	The HTTP proxy server name or IP address
Proxy port	The HTTP proxy port to use (usually 8080)

Option	Description
Ignore proxy for hosts	Specify a regular expression matching the hosts you want to ignore, | separated. For example 127\.0\..*

13.2.14. Create a file

Icon



Create file

13.2.14.1. General description

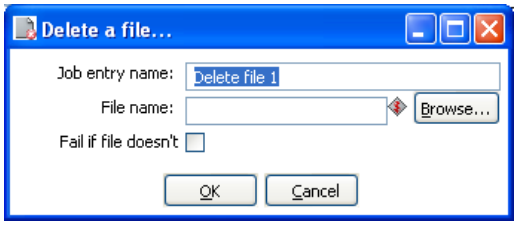
You can use the Create a file job entry to create an empty file. This is useful for creating "trigger" files from within jobs.

13.2.14.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job.
File name	The name and path of the file to create an empty file.
Fail if file exists	The job entry will follow the failure outgoing hop when the file to be created already exists (empty or not) and this option is switched on. The default is on.

13.2.15. Delete a file

Icon



Delete file

13.2.15.1. General description

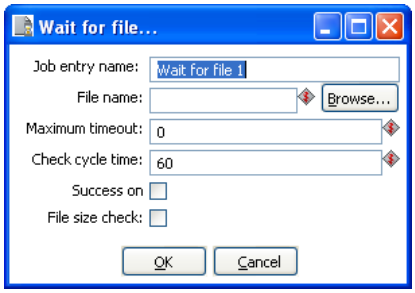
You can use the Delete a file job entry to delete a file (empty or not).

13.2.15.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job.
File name	The name and path of the file to delete.
Fail if file doesn't exist	The job entry will follow the failure outgoing hop when the file to be deleted does not exist anymore and this option is switched on. The default is off.

13.2.16. Wait for a file

Icon



Wait for file

13.2.16.1. General description

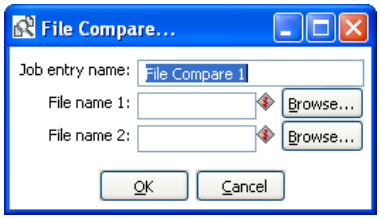
You can use the Wait for file job entry to wait for a file. This job entry will sleep and periodically check whether the specified file exists after which the flow will continue. The job entry can either wait indefinitely for the file or it can timeout after a certain time.

13.2.16.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job.
File name	The name and path of the file to wait for.
Maximum timeout	The maximum timeout in number of seconds, or 0 to wait indefinitely. This is the number seconds after which the flow will continue even if the file was not created. When the timeout is reached the "Success on timeout" option will determine whether the outgoing success or failure hop will be followed.
Check cycle time	The time in seconds between checking for the file. The file will be checked for in the start of the execution and then every "check cycle time" seconds until the maximum timeout is reached. A job can only be stopped every "check cycle time" as else the job entry step will be sleeping. A check cycle time of 30 or 60 seconds seems to be a good trade-off between the period until the file is detected and the required CPU usage.
Success on timeout	This option determines what to do when the "Maximum timeout" has been reached. If enabled, the job entry will evaluate to success the success outgoing hop will be followed if the file is not detected after timeout.
File size check	When this is switched on the job entry will after detecting the specified file, only continue if the file size hasn't changed the last check "cycle time seconds". This is useful e.g. if a file is created in place (although it's recommended to generate a file elsewhere and then move it in place).

13.2.17. File compare

Icon



File Compare

13.2.17.1. General description

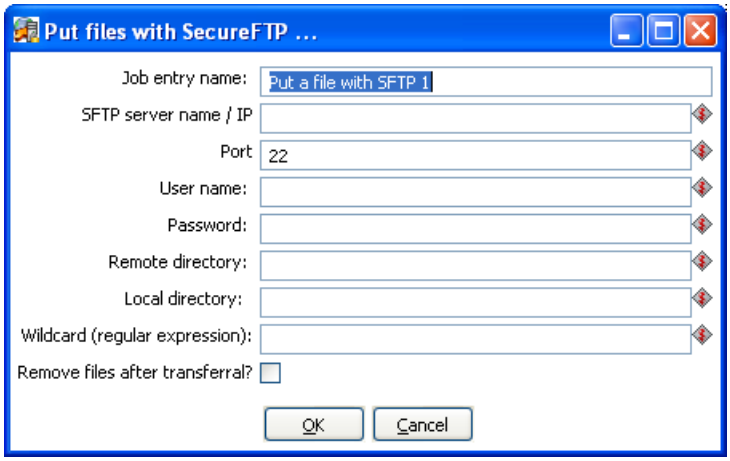
You can use the File compare job entry to compare the contents of 2 files and control the flow of the job by it. When the contents of the files are the same the success outgoing hop will be followed, else the failure hop will be followed.

13.2.17.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job.
File name 1	The name and path of the file of the first file to compare.
File name 2	The name and path of the file of the second file to compare.

13.2.18. Put a file with SFTP

Icon



File Compare

13.2.18.1. General description

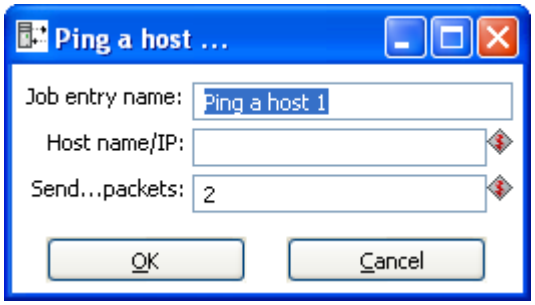
You can use the Put files with SFTP job entry to put one or more files to an FTP server using the Secure FTP protocol.

13.2.18.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
SFTP-server name (IP)	The name of the SFTP server or the IP address
SFTP port	The TCP port to use. This is usually 22
User name	The user name to log into the SFTP server
Password	The password to log into the SFTP server
Remote directory	The remote directory on the SFTP server to which we put the files
Local directory	The directory on the machine on which Kettle runs from which you want to ftp the files from
Wildcard	Specify a regular expression here if you want to select multiple files. For example: `.*txt$ : get all text files` `A.*[0-9]\.txt : files tarting with A ending with a number and .txt`
Remove files after transferral?	Remove the files after they have been successfully transferred.

13.2.19. Ping a host

Icon



Ping a host

13.2.19.1. General description

You can use the Ping a host job entry to ping a host using the ICMP protocol.

13.2.19.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Host name/IP	The name or the IP address of the host to ping
Send...packets	The number of packets to send (by default 2)

13.2.20. Wait for

Icon

Wait for

13.2.20.1. General description

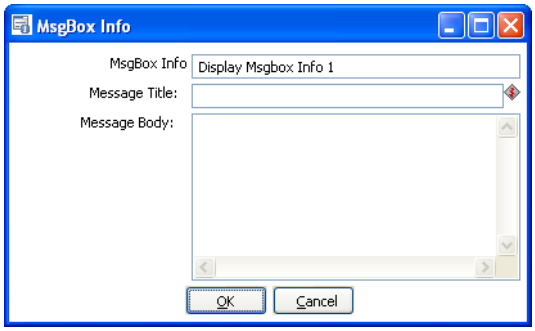
You can use the Wait for to wait a delay before running the next job entry.

13.2.20.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Wait for	The delay to wait
Unit time	Specify the unit time (second, minute and hour)

13.2.21. Display Msgbox info

Icon



MsgBox Info

13.2.21.1. General description

This job entry allow you to display a message box in job. You can easily see where you are in the process. This entry is only available using the Graphical User Interface to execute the job.

13.2.21.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Message title	The title of the message
Message body	The message to display

13.2.22. Abort job

Icon

MsgBox Info

13.2.22.1. General description

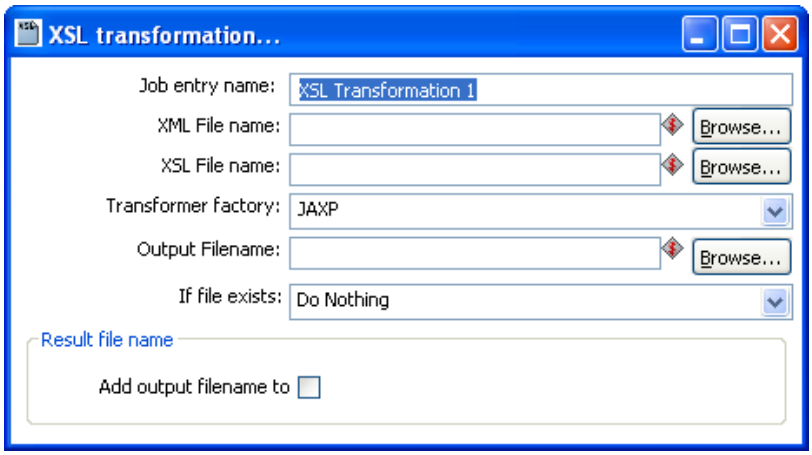
Use this job entry if you want to abort a job.

13.2.22.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Message	Message to add in log when aborting

13.2.23. XSL transformation

Icon



XSL Transformation

13.2.23.1. General description

XSL transformation job entry is designed to transform (by applying XSL document) XML documents into other documents (XML or other format, such as HTML or plain text).

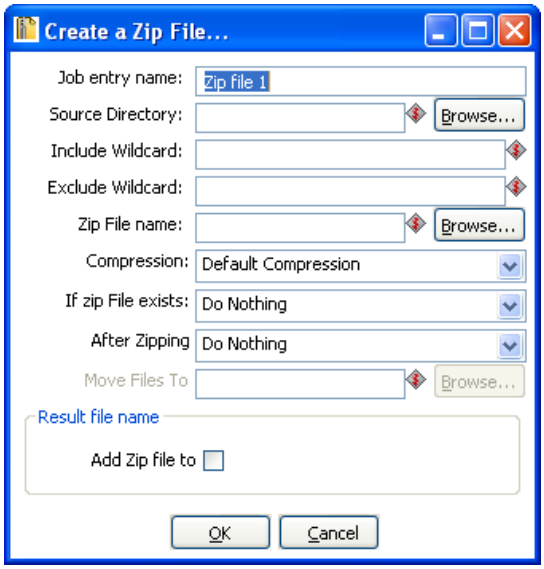
The original document is not changed; rather, a new document is created based on the content of the XML file.

13.2.23.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
XML File name	The full name of the source XML file
XSL File name	The full name of the XSL file
Output File name	The full name of the created document (result of XSL transformation)
If file exists	Define the behavior when an output file with the same name exists Options : – Create new with unique name : a new output file will be created – Do nothing : nothing will be done – Fail : the job will fail

13.2.24. Zip files

Icon



Create a Zip file

13.2.24.1. General description

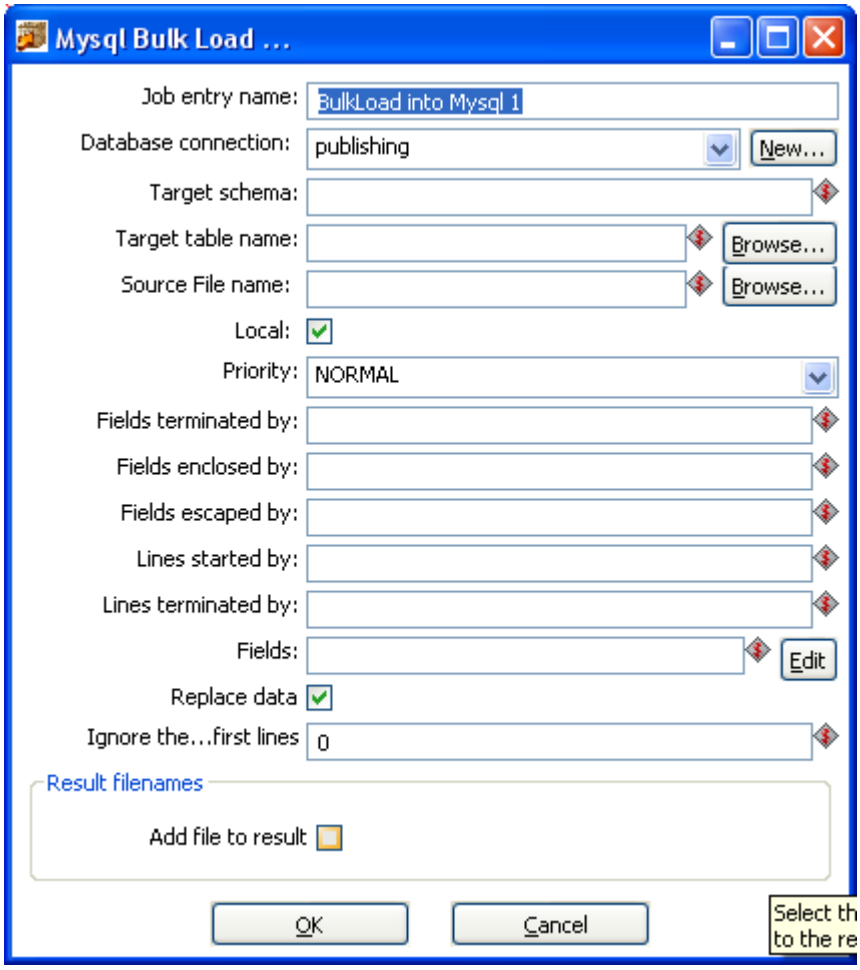
This step creates a standard ZIP archive using the options you specify in the dialog.

13.2.24.2. Options

Option	Description
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Source directory	The source directory of the files to be zipped
Include wildcard	The wildcard (regular expression) of the files to include in the zip archive
Exclude wildcard	The wildcard (regular expression) of the files to exclude from the zip archive
Zip file name	The full name of the destination archive
Compression	The compression level to be used (Default, Best Compression, Best speed)
If zip file exists	The action to take when there already is a file at the target destination.
After zipping	The action to take after zipping
Move files to	The target directory to move the source files to after zipping

13.2.25. Bulkload into MySQL

Icon



Bulkload into MySQL

13.2.25.1. General description

This step is used to perform bulk load operations from a flat file into a MySQL database table.

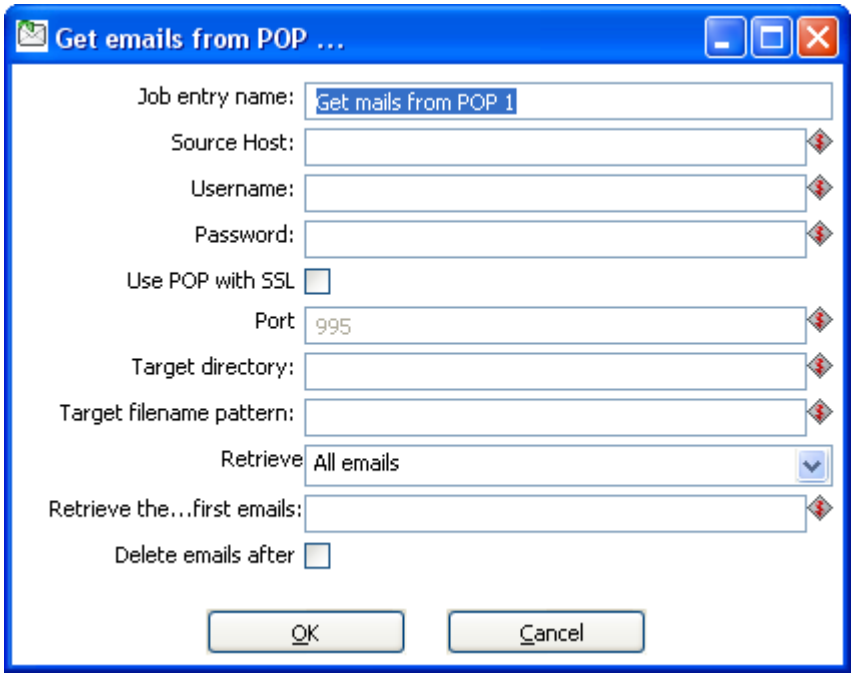
13.2.25.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Database connection	The database connection used to write data to.
Target schema	The name of the Schema for the table to write data to. This is important for data sources that allow for table names with dots '.' In it.
Target table name	The name of the table to write data to.
Source file name	Name of the text file to load data from.
Local	Enabled: the file is read by the client program on the client host and sent to the server. Disabled: the file must be located on the server host and is read

Option	Description
	directly by the server.
Priority	Specify the priority in MySQL for the bulk load.
Fields terminated by	Specify the fields delimiter in the text file source.
Fields enclosed by	Specify the enclosure character for fields in the source text file.
Fields escaped by	Specify the escape character for fields in the source text file.
Lines started by	Specify the character(s) used to indicate the start of a row in the source text file.
Lines terminated by	Specify the character(s) used to indicate the end of a row in the source text file.
Fields	Specify the names of attributes of <tableName> that are set by your data file (separated by commas). Any attributes unspecified in the list of attributes will be set to NULL.
Replace data	Enable this option to overwrite existing data in the target table.
Ignore the first ... lines	Optionally specify a number of lines to ignore.
Add files to result	Enable this to add the destination files to the results file names. This is useful if you want to attach these files to an email using the Email job entry.

13.2.26. Get Mails from POP

Icon



Get Mails from POP

13.2.26.1. General description

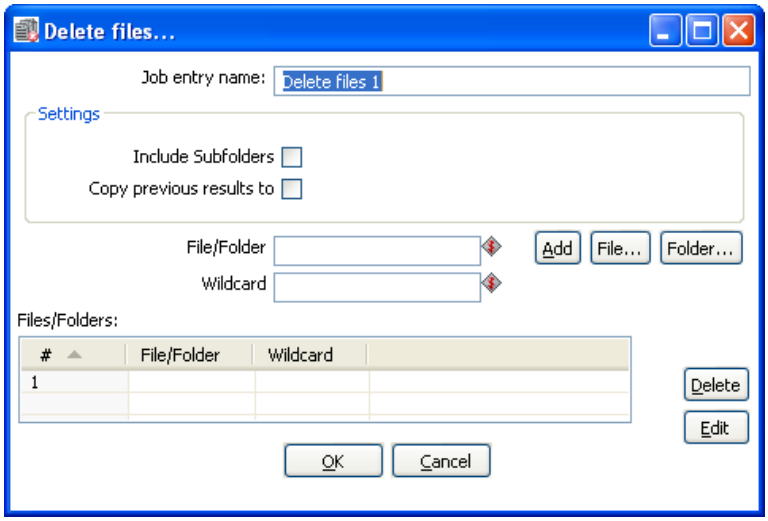
This step provides the ability to read one or more emails from a POP server.

13.2.26.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Source Host	The host name or IP address of the POP mail server.
Username	The username for authenticating to the POP server.
Password	The password for authenticating to the POP server.
Use POP with SSL	Enable this option to connect using a Secure Socket Layer (SSL) connection.
Port	When SSL option is enabled, use this property to set the IP port for SSL communication with the POP server.
Target directory	Specify the target directory for where to land the emails retrieved.
Target filename pattern	Specify the regular expression wildcard used to identify the target filenames.
Retrieve	Use this to specify whether to retrieve all emails, unread emails, or a specific number of emails.
Retrieve the .. first emails	If the Retrieve property is set to 'First...emails', this property is used to specify the number of emails to retrieve.
Delete emails after	Enable this option to delete all retrieved emails from the POP server.

13.2.27. Delete Files

Icon



Delete Files

13.2.27.1. General description

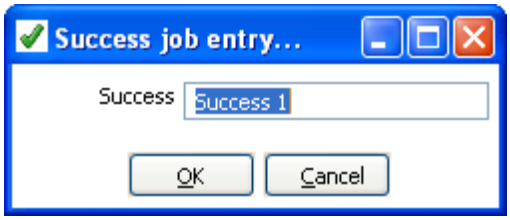
This step is used to delete one or more files from a specified folder.

13.2.27.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Include Subfolders	Enable this option to also delete matched files from subfolders of the target directory.
Copy previous results to args?	Enable this to pass the results of the previous entry to the arguments of this entry.
File/Folder	The target file or folder to delete files from.
Wildcard	The regular expression used to define the file name pattern for the files to delete.
Files/Folders Table	This table displays the list of currently defined files and folders to delete.

13.2.28. Success

Icon



Success

13.2.28.1. General description

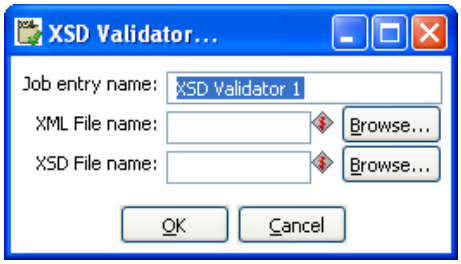
This step is similar to the dummy step in that it does not do anything. Its main function is as a placeholder for routing the job flow upon successful evaluation.

13.2.28.2. Options

Option	Description
Success	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.

13.2.29. XSD Validator

Icon



XSD Validator

13.2.29.1. General description

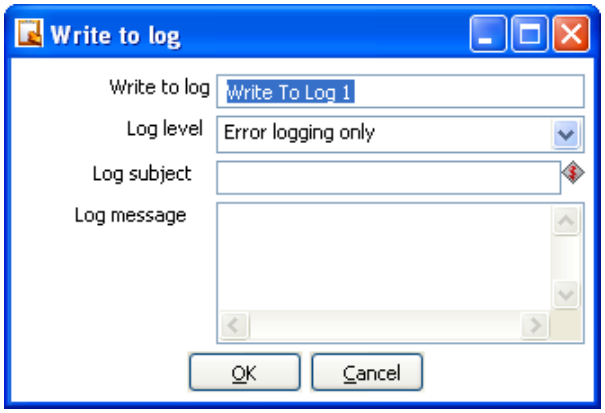
This step will validate an XML file against and XML Schema Definition (XSD).

13.2.29.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
XML File name	Specify the name of the XML document to validate.
XSD File name	Specify the name of the XSD file used for validation of the XML document.

13.2.30. Write to log

Icon



Write to log

13.2.30.1. General description

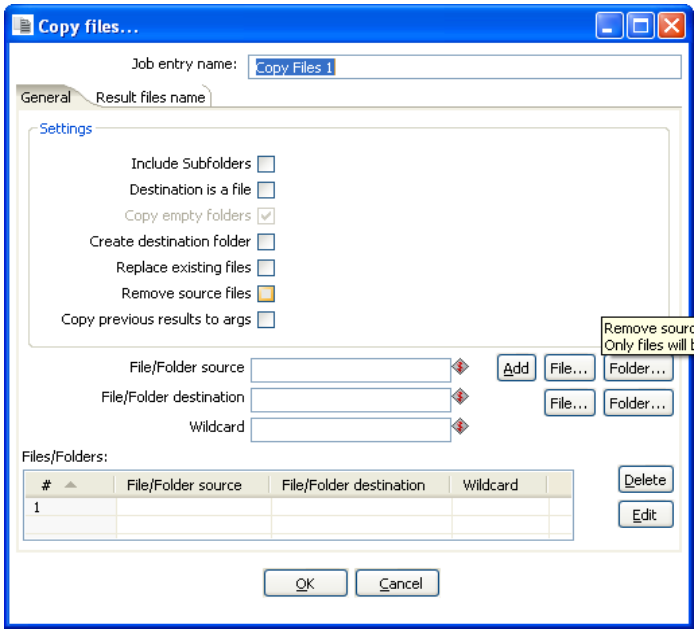
This step provides the ability to write an entry to the execution log.

13.2.30.2. Options

Option	Description
Write to log	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Log level	Specify the log level condition for when the specified log message should be written to the log.
Log subject	Specify a short subject for the log message.
Log message	Specify the detailed message to be written to the log file.

13.2.31. Copy Files

Icon



Copy Files

13.2.31.1. General description

This step provides the ability to copy one or more files to another location.

13.2.31.2. General Tab

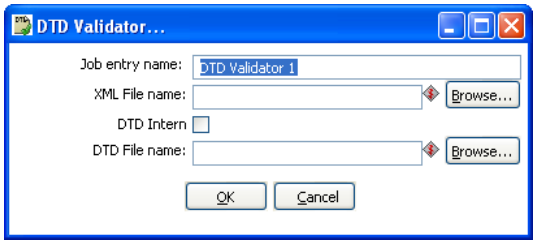
Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Include Subfolders	Enable this option to also copy matched files from subfolders of the target directory.
Destination is a file	Enable this option if the target of the copy is a file.
Copy empty folders	If including subfolders, this option allows you to specify whether or not to copy empty folders.
Replace existing files	Enable this option to automatically overwrite any existing files.
Remove source files	Enable this option to remove the source files after copy is completed.
Copy previous results to args	Enable this to pass the results of the previous entry to the arguments of this entry.
File/Folder source	Specify the source file or folder to copy.
File/Folder destination	Specify the target file or folder to copy files to.
Wildcard	The regular expression used to define the file name pattern for the files to copy.
File/Folders	This table displays the list of currently defined files and folders to copy.

13.2.31.3. Results files names

The 'Add files to result files name' will add the destination files to the results file names. This is useful if you want to attach these files to an email using the Email job entry.

13.2.32. DTD Validator

Icon



DTD Validator

13.2.32.1. General description

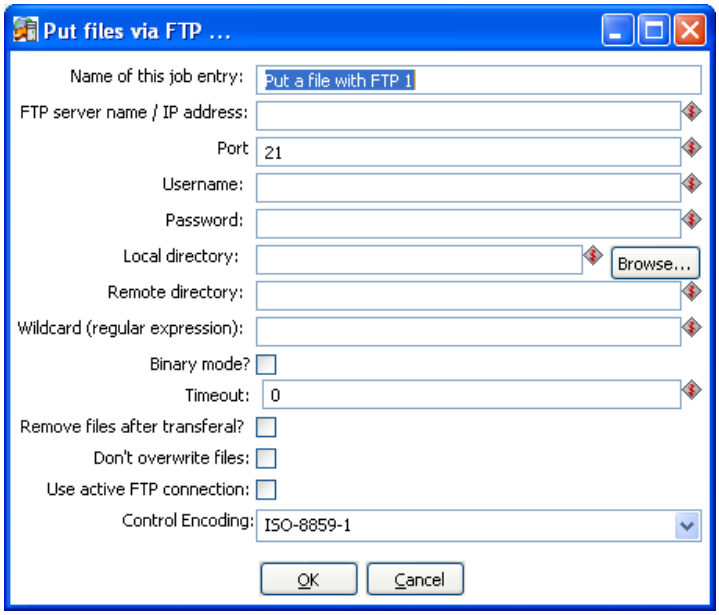
This step provides the ability to validate an XML document against a Document Type Definition (DTD).

13.2.32.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
XML File name	Specify the XML document to validate.
DTD Intern	Enable this option if the DTD is defined withing the XML document being validated.
DTD File name	Specify the file name containing the DTD used for validation.

13.2.33. Put a file with FTP

Icon



Put files with FTP

13.2.33.1. General description

You can use the Put files with FTP job entry to put one or more files on an FTP server using the FTP protocol.

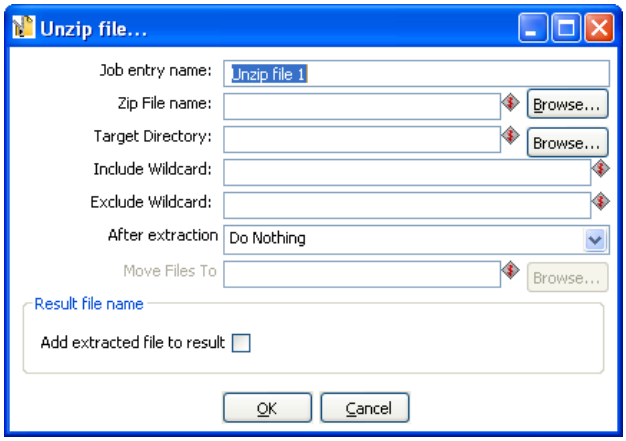
13.2.33.2. Options

Option	Description
Name of this job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
FTP server name/IP address	The name of the FTP server or the IP address
Port	The TCP port to use. This is usually 21
Username	The user name to log into the FTP server
Password	The password to log into the FTP server
Local directory	The directory on the machine on which Kettle runs from which you want to FTP the files from
Remote directory	The remote directory on the FTP server to which we put the files
Wildcard (regular expression)	Specify a regular expression here if you want to select multiple files. For example: `.*txt$ : get all text files` `A.*[0-9]\.txt : files tarting with A ending with a number and .txt`
Binary mode?	Enable this option to perform the transfer in Binary mode.

Option	Description
Timeout	Specify the timeout period before ending in error.
Remove files after transferal?	Remove the files after they have been successfully transferred.
Don't overrite files	Enable this option to prevent overwriting any existing files on the target FTP server.
Use active FTP connection	Enable this option to use and active FTP connection.
Control Encoding	Specify the character set to user for filenames and directories.

13.2.34. Unzip

Icon



Unzip

13.2.34.1. General description

This step is used to decompress a zip file to a specified location.

13.2.34.2. Options

Option	Description
Job entry name	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Zip File name	Specify the name of the file to unzip.
Target Directory	Specify the target directory to place the unzipped contents.
Include Wildcard	Specify a regular expression wildcard for the specific files you want to unzip.
Exclude Wildcard	Specify a regular expression wildcard for any files you want to exclude from the unzip process.
After extraction	Specify the action to take after unzipping the file. Options include do nothing, delete files, or move files to specified location.
Move Files To	If the move files action is specified in the 'After Extraction' property, this field is used to identify the target location to move the files to.
Add extracted file to result	Enable this to add the destination files to the results file names. This is useful if you want to attach these files to an email using the Email job entry.

13.2.35. Dummy Job Entry

Icon

13.2.35.1. General description

This is an example plugin used to illustrate how to develop and deploy your own custom plugins into Kettle. The functionality of the Dummy Job Entry (Get files from SFTP) is arbitrary and only provided to highlight a working plugin. For more information about developing Kettle plugins, visit Writing your own Pentaho Data Integration Plug-In on the Pentaho Wiki.

14. Graphical View

14.1. Description

The Graphical View tab contains the canvas on which transformations and jobs are drawn. There will be a separate tab for each job and/or transformation you currently have open with an icon indicating the file type:

Transformation Icon

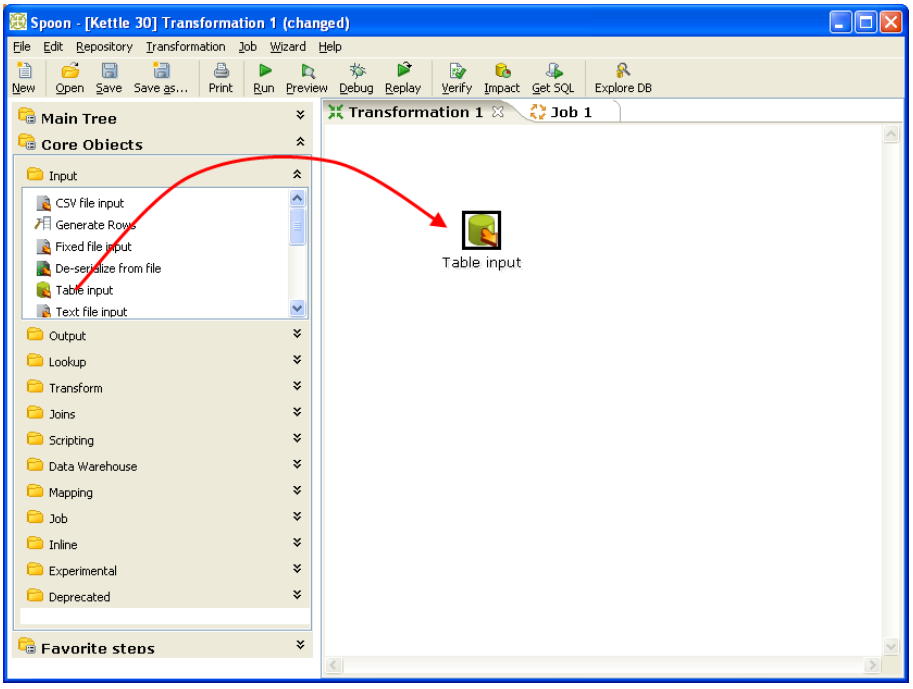
Job Icon

The Graphical View tab(s) provide an easy to understand representation of the work that needs to be done and the flow of the data.

14.2. Adding steps or job entries

14.2.1. Create steps by drag and drop

Adding steps to a transformation (or a job entry to a job) on the canvas is easy: simply select a step type from the tree on the left and drag in onto the canvas:



At the location of the mouse you will see a square that represents the location of the steps when you let go of the button. When you let go of the mouse button the selected step (Text file input) will become part of the transformation.

You can also add a transformation step by right-clicking on the workspace and selecting New Step...|*Step type.*

14.3. Hiding a step

If you right click on a step or job entry that is drawn on the graphical view, you will get a popup-menu that allows you to select the option: "Hide step". This will remove the step from the graphical view, but not delete it.

14.4. Transformation Step options (right-click menu)

This section describes the right-click menu options when you right-click on a transformation step in the Graphical View.

14.4.1. Edit step

This opens the step dialog so that you can change its settings.

14.4.2. Edit step description

This opens a dialog that allows you to enter a textual description of the step.

14.4.3. Data movement

See Distribute or copy for a complete description of the available data movement options.

14.4.4. Change number of copies to start...

See Launching several copies of a step.

14.4.5. Copy to clipboard

This option allows you to copy the XML defining the step to the clipboard. You can then paste this step into another transformation.

14.4.6. Duplicate Step

This option will create a copy, positioned a bit lower to the right of the original step.

14.4.7. Delete step

This will permanently remove the step from the transformation.

14.4.8. Hide Step

This will hide the step in the Graphical View, but not remove it from the transformation.

14.4.9. Show input fields

This option tries to determine all the fields and their origin by tracing the input-streams back to their source.

14.4.10. Show output fields

This option adds the fields of the current step to the ones of the input fields and shows the result.

14.5. Job entry options (right-click menu)

14.5.1. Open Transformation/Job

This opens a new tab displaying the selected transformation or job.

14.5.2. Edit job entry

This opens the dialog for the job entry allow you to change the settings

14.5.3. Edit job entry description

This opens a dialog that allows you to enter a textual description of the job entry.

14.5.4. Create shadow copy of job entry

This option will create a copy, positioned a bit lower to the right of the original job entry.

14.5.5. Copy selected entries to clipboard (CTRL-C)

Copies the XML describing the selected job entries to the clipboard.

14.5.6. Align / distribute

This option allows you to keep the graph clean by aligning job entries with each other.

14.5.7. Detach entry

Unlinks this job entry from the hops that connect it to other steps.

14.5.8. Delete all copies of this entry.

Delete all copies of this job entry, not just this one!

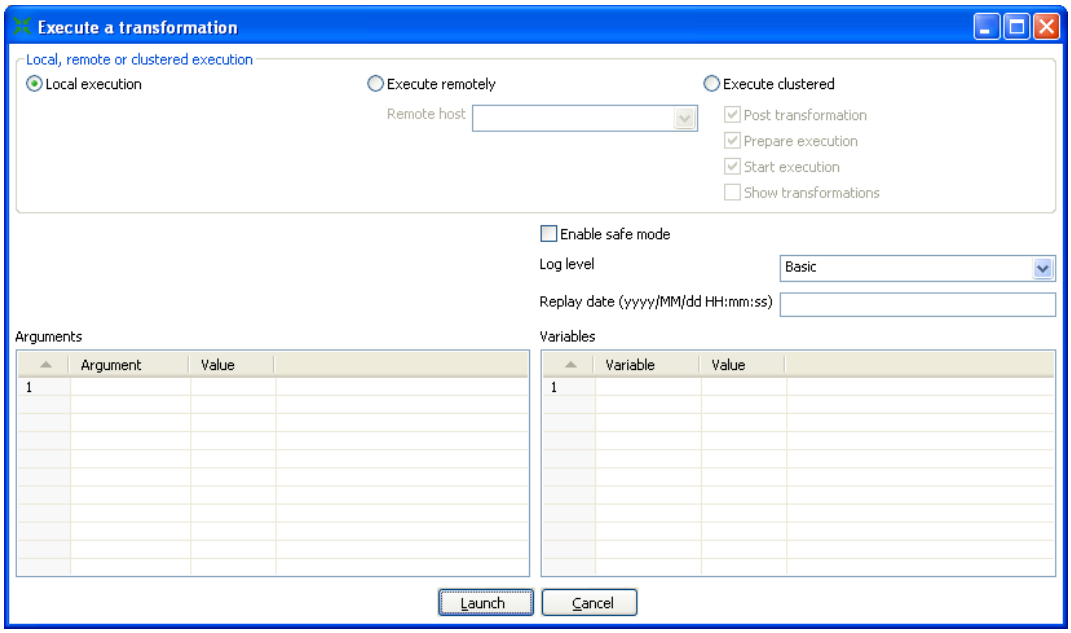
14.6. Adding hops

On the graphical view the quickest way to create a new hop is by dragging with the mouse from one step to another using the middle button. You can also drag the left button and press the SHIFT key at the same time. For a more complete explanation regarding hops, please refer to chapter on Hops.

15. Running a Transformation

15.1. Running a Transformation Overview

When you are finished modifying your transformation, you can run it by clicking on the run button from the main menu, toolbar or by pressing F9.



Execute a transformation

15.2. Execution Options

15.2.1. Where to Execute

There are three options for deciding where you want your transformation to be executed:

- **Local Execution**: the transformation or job will be run on the machine you are currently using
- **Execute remotely**: allows you to specify a remote server where you want the execution to take place. This feature requires that you have Pentaho Data Integration installed on a remote machine and running the Carte service. See the 14.4.3 Configuring a remote or slave server for more details on setting up remote and slave servers.
- **Execute clustered**: Allows you to execute the job or transformation in a clustered environment. See the section on Clustering for more details on how to execute a job or transformation in a clustered environment.

15.2.2. Other Options

The following table provides a detailed description of other Execution options:

Option	Description
Enable Safe mode	Places the transformation in Safe Mode. Additional row checking is enabled at runtime, see also: Safe Mode
Log level	This allows you to specify the level of detail you want to capture in the log. For

Option	Description
	detailed descriptions of the log level types see Logging.
Replay date	This will set the replay date for when you want to replay the transformation. It will pick up information in the text file input or Excel input steps to skip rows already processed on the replay date.
Arguments	This grid allows you to set the value of arguments to be used when running the transformation.
Variables	This grid allows you to set the value of variables to be used when running the transformation.

15.3. Setting up Remote and Slave Servers

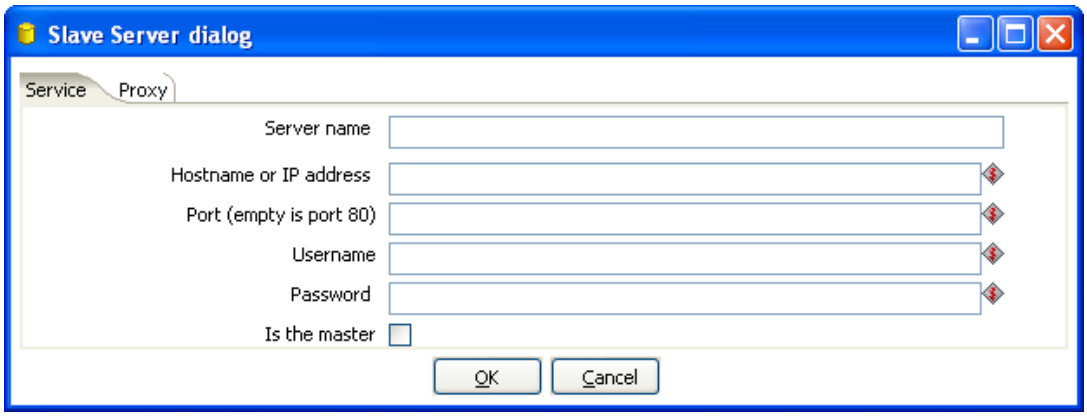
15.3.1. General description

Slave servers allow you to execute a transformation on a remote server. Setting up a slave server requires having a small web-server running on your remote machine called "Carte" that will accept input from either Spoon (remote & clustered execution) or from the Transformation job entry (clustered execution).

15.3.2. Configuring a remote or slave server

Install Pentaho Data Integration on server you want to use to remotely execute transformations (for more information on setting up a remote server, see the chapter on Installation). The installation includes a small web server called Carte used to support remote requests. Start the Carte server by running Carte.bat (Windows) or carte.sh from the root of your Pentaho Data Integration installation.

Next, you need to point your master server to each of the slave server. To do this, double click on 'Slave server' node in the tree control on the left, or by right-clicking on 'Slave Server' and selecting the New Slave Server option.



15.3.2.1. Service tab options

Option	Description
Server name	The friendly name of the server you wish to use as a slave
Hostname or IP address	The address of the machine to be used as a slave
Port	Defines the port you wish to use for communicating with the remote server
Username	Enter the username credential for accessing the remote server

Option	Description
Password	Enter the password credential for accessing the remote server
Is the master	This setting tells Pentaho Data Integration that this server will act as the master server in any clustered executions of the transformation

Note: *when executing a transformation in a clustered environment, you should have 1 server setup as the master and all remaining servers in the cluster as slaves.*

15.3.2.2. Proxy tab options

Option	Description
Proxy server hostname	Sets the hostname for the Proxy server you are connecting through
The proxy server port	Sets the port number used in communication with the proxy
Ignor proxy for hosts: regexp|separated	Specify the server(s) for which the proxy should not be active. This option supports specifying multiple servers using regular expressions. You can also add multiple servers and expressions separated by the '|' character.

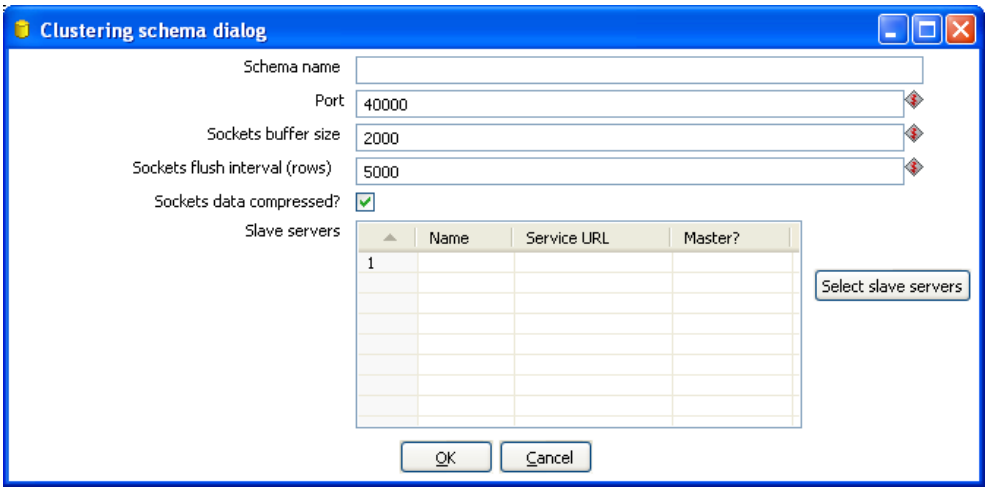
15.4. Clustering

15.4.1. Overview

Clustering allows transformations and transformation steps to be executed in parallel on more than one server. The clustering schema defines which slave servers you want to assign to the cluster and a variety of clustered execution options.

15.4.2. Creating a cluster schema

Begin by double-clicking on the 'Kettle cluster schemas' node in the tree on the left or right-clicking on that node and selecting 'New clustering schema':



15.4.3. Options

Option	Description
Schema name	The name of the clustering schema
Port	Here you can specify the port from which to start numbering ports for the slave servers. Each additional clustered step executing on a slave server will consume an additional port. **Note**: *Make sure no other networking protocols are in the same range to avoid networking problems.*
Sockets buffer size	The internal buffer size to use.
Sockets flush interval	The amount of rows after which the internal buffer is sent completely over the network and emptied.
Sockets data compressed?	When this is checked, all data is compressed using the Gzip compression algorithm to minimize network traffic.
Slave Servers	This is a list of the servers to be used in the cluster. You should have one master server and any number of slave servers. To add servers to the cluster, click on the 'Select slave servers' button to select from the list of available slave servers. See (for more details on how to create a slave server.

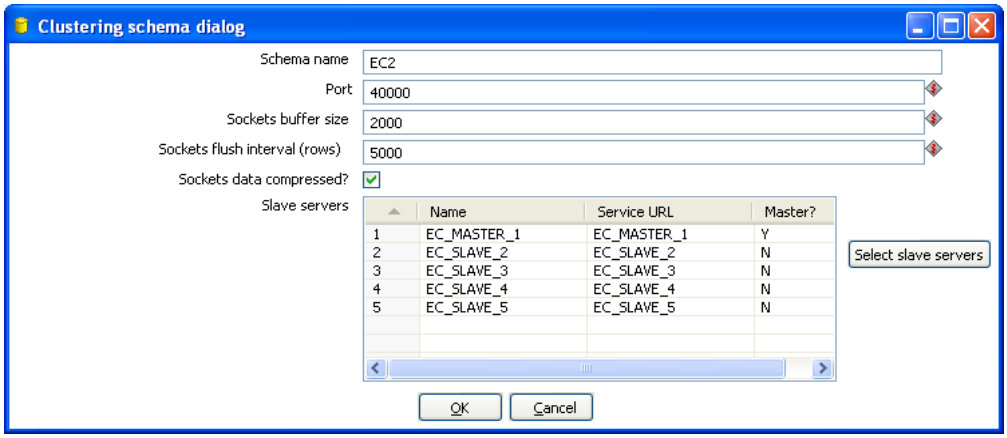
15.4.4. Running transformations using a cluster

When you chose to run a Transformation, select the 'Execute clustered'. You will have the following options:

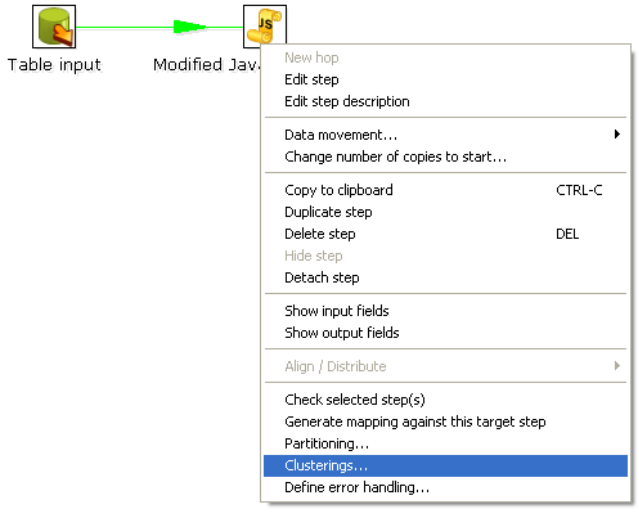
- **Post transformation**: Split the transformation and post it to the different master and slave servers.
- **Prepare execution**: This runs the initialization phase of the transformation on the master and slave servers.
- **Start execution**: This starts the actual execution of the master and slave transformations.
- **Show transformations**: Show the generated (converted) transformations that will be executed on the cluster (see the Basic Clustering Example for more information generated transformations).

15.4.5. Basic Clustering Example

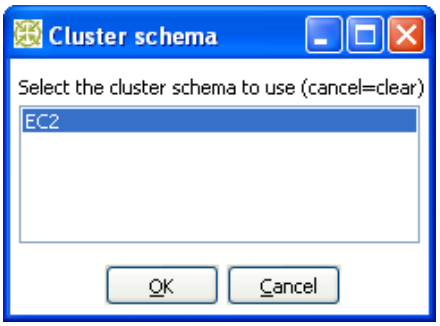
Suppose that you have data from a database table and you want to run it through a particularly complex JavaScript program. For performance reasons you want to execute this program on 5 different hosts. You begin by creating a cluster with one master server and 4 slaves:



Then you create the transformation as usual, connecting 2 steps with a hop. You specify that the script to execute is running on a cluster:



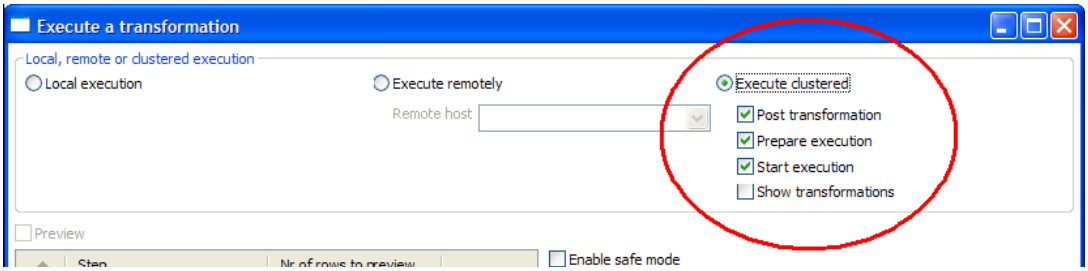
Then select the cluster to use:



The transformation will then be drawn like this on the graphical view:

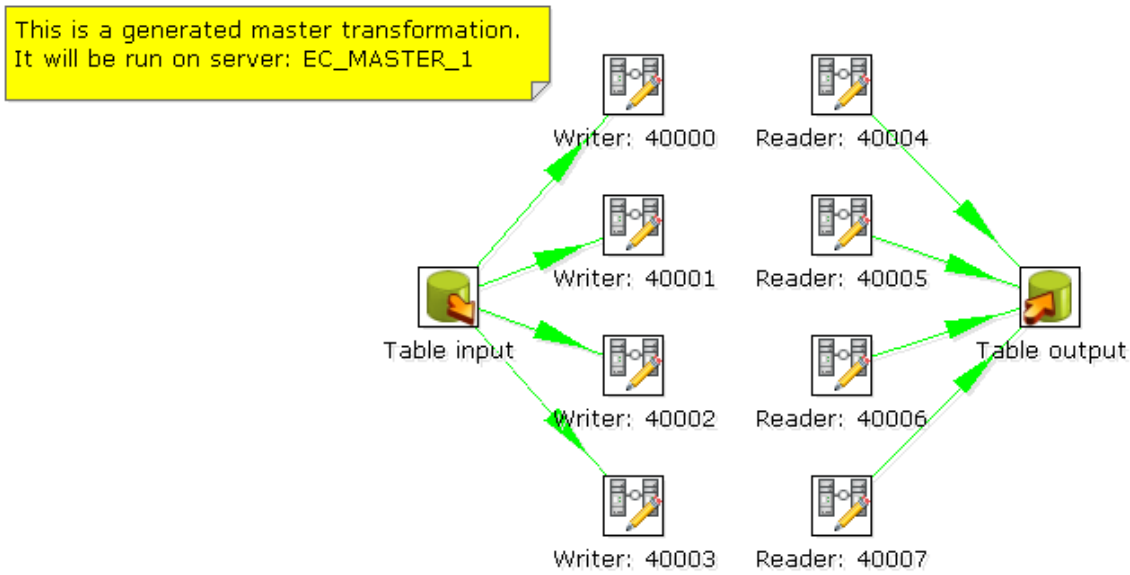
The Cx4 indicates that the step will be executed on a cluster. Suppose we then store the calculated information as a result in a table again:

When we execute this transformation locally, we will see no difference with the usual result you expect from a non-clustered execution. That means that you can use the normal local execution to test the transformation. However when we can execute the transformation in a clustered fashion like this:

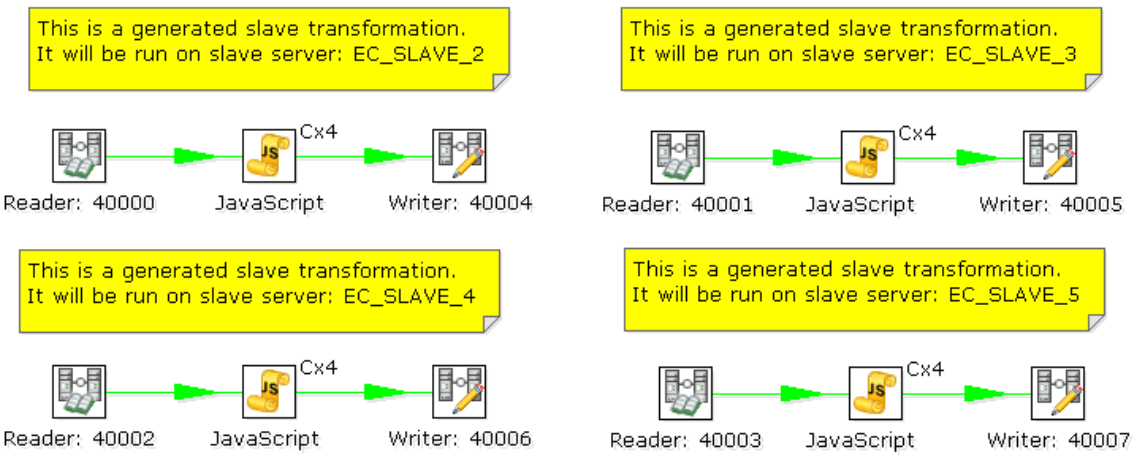


In this case, 5 transformations will be generated for the 5 servers in the cluster.

One master:



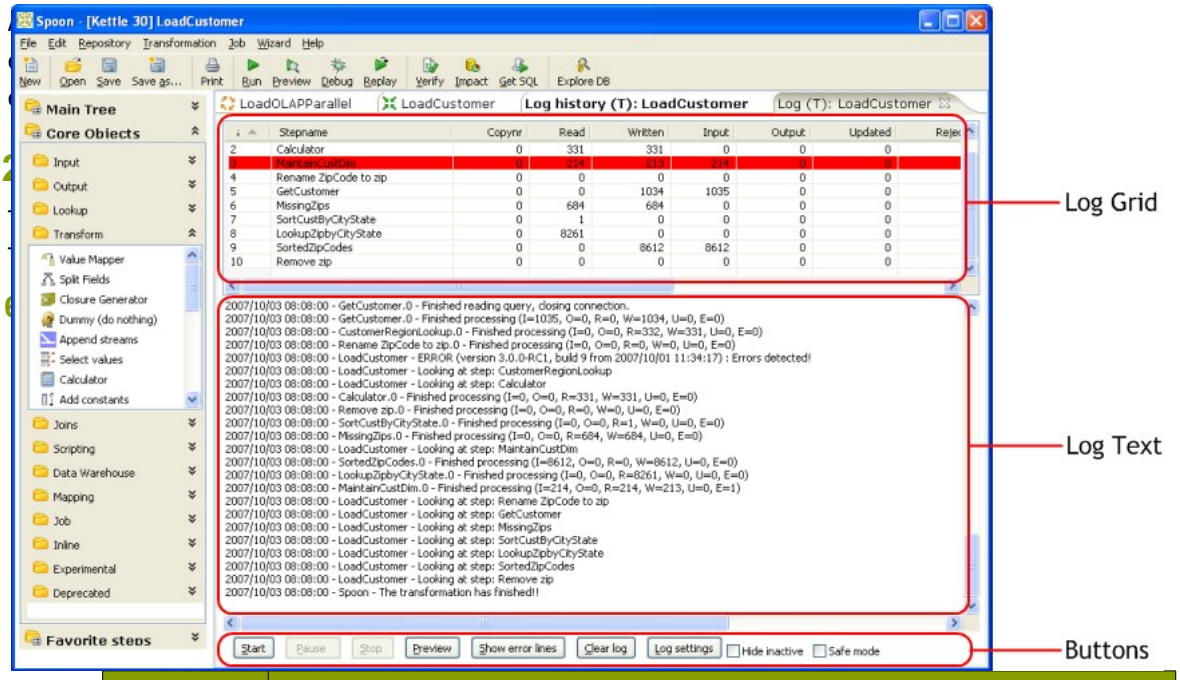
And 4 slaves transformations:



As you can see, data will be sent over the TCP/IP sockets using the Socket Writer and Socket Reader steps.

16. Logging

16.1. Logging Description



Option	Description
Stepname	The name of the step
Copynr	Copy number of the step
Read	Number of lines read from input-streams
Written	Number of lines written to output-streams
Input	Number of lines read from file or database
Output	Number of lines written to file or database
Updated	Number of lines updated in the database
Rejected	Number of errors that occurred
Errors	The status of the step: running, finished or stopped
Active	The number of seconds that the step has been running.
Time	The speed in rows per second at which the step processes rows.
Speed	Priority of the step (10=highest, 1=lowest), nr of rows in the input-stream(s), nr of rows in the output-stream(s).
input/output	Sleep time (get/put) is the time that a step had to go to sleep (in nano seconds) because the input buffer was empty (get) or the output buffer was full (put).

NOTE: *The system is tuning the steps priority in such a way that the slowest steps get the highest priority.*

16.2.2. Job Log Grid

The log grid displays the following details for each job entry executing in the Job:

Option	Description
Job/Job Entry	The name of the job / job entry
Comment	A comment on the state of the entry execution
Result	The result (success or failure) of the job entry
Reason	Reason: why was this job entry started?
Nr	The value of the nr variable in the result object (available in evaluation Javascript)
Log date	Log date: logging date, corresponds with the start or end of the job entry.

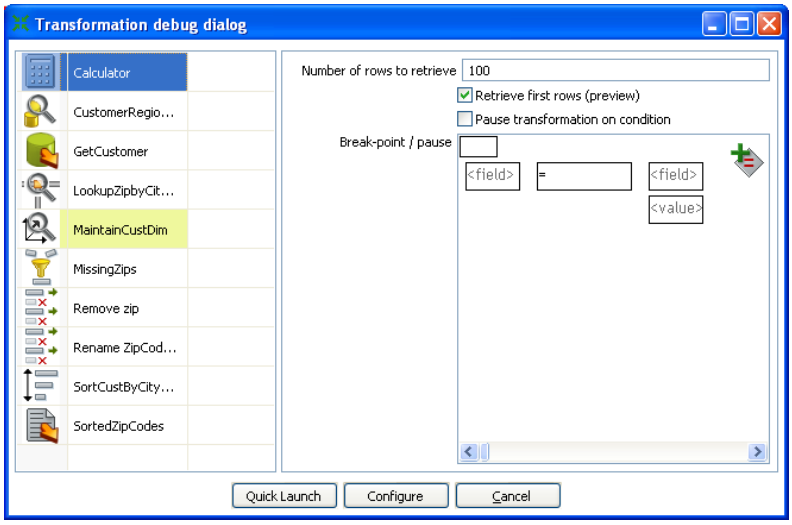
16.3. Buttons

16.3.1. 15.4.1 Transformation Buttons

16.3.1.1. 15.4.1.1. Start

This button starts the transformation. Please note that Spoon tries to launch this from the XML-file or repository. It is therefore necessary that the transformation is saved. The output of the execution is displayed in the Log Text part of the Log View.

16.3.1.2. Preview (debug)



This button launches the Transformation Debug dialog allowing you to specify the number of rows to preview and define conditional breakpoints for the preview execution. After configuring the debug information, click the 'Quick Launch' button to begin the preview execution for the currently selected step. The output of the execution is displayed in the Log Text part of the Log View.

16.3.1.2.1. Debug Options

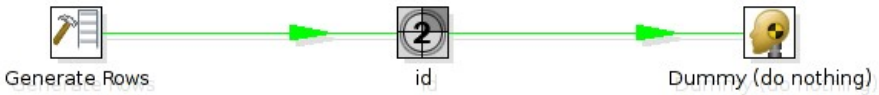
The following table provides a detailed description of the debug options:

Option	Description
Step List	The step list on the left displays a list of available steps from the current transformation. Select a step to begin configuring related

Option	Description
	options like number of rows and break-points.
Number of rows to retrieve	Enter the rows per step you want to preview for the selected step. After the requested rows are obtained from the different steps, the transformation is ended and the result is shown. **Note:** *This option will only take effect when the 'Retrieve first rows' option is checked.*
Retrieve first rows (preview)	Enable this to restrict the preview size to the number of rows specified above.
Pause transformation on condition	Enable this option to cause the transformation to pause if one of the conditional break-points evaluates to true during execution.
Break-point/pause condition	Enter conditions based on comparing one field to another field or value.

16.3.1.2.2. Debug example

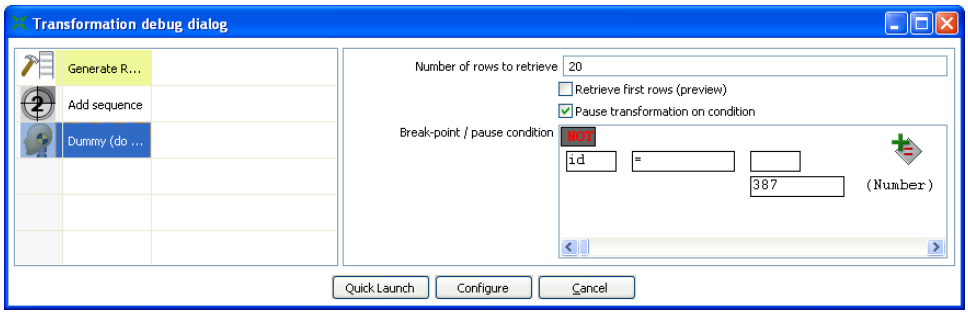
Starting with the simple transformation shown here:



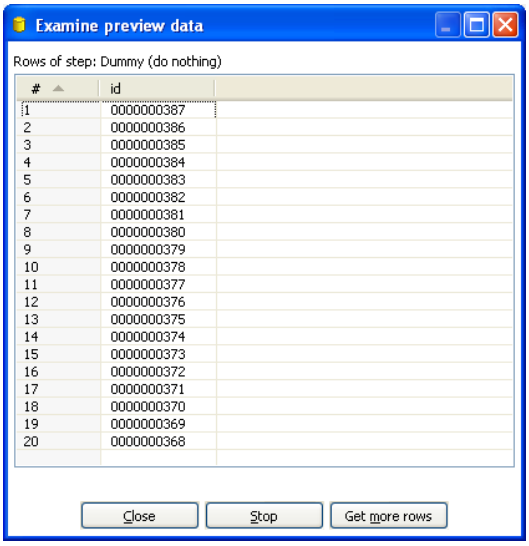
The generate rows step generates empty rows and adds an id from 1 to 1000. Now we want to pause the transformation and see the content of the row where:

```
id=387
```

To do this, simply click on the debug icon in the main toolbar:



As you can see, we can specify a condition on which the transformation is paused. You can also specify to keep the last *N* rows in memory *before* the condition was met. Pressing the 'Quick Launch' button will begin the preview execution and display the following dialog upon meeting the condition:



For convenience, the order of the rows is reversed in the preview window so that the row that met the condition is always at the top of the results. After closing the preview window, you will note that the transformation is paused (see log tab) and you can then resume execution by clicking the resume button. If a condition is met again, the transformation will be paused again and another preview dialog will display.

16.3.1.3. Show error lines

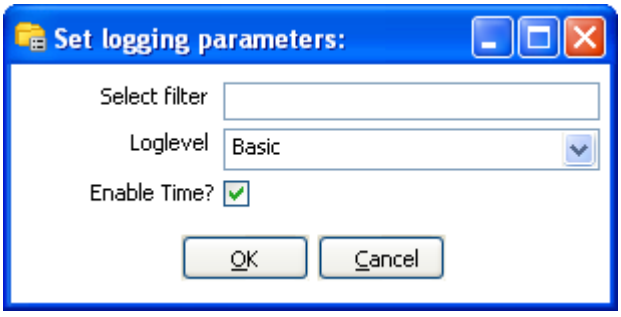
This button displays all lines of the Log Text that contain the word ERROR (lower- or uppercase). You can then choose to edit the source step of the error.

16.3.1.4. Clear log

This clears the text in the Log Text Window.

16.3.1.5. Log Settings

This is the "Log Settings" dialog:



If you put a text in the filter field, only the lines that contain this text will be shown in the Log Text window.

The "Log level" setting allows you to select the logging level. You can choose one of these:

- Nothing: Don't show any output
- Error: Only show errors
- Minimal: Only use minimal logging
- Basic: This is the default basic logging level
- Detailed: Give detailed logging output
- Debug: For debugging purposes, very detailed output.

Row level: Logging at a row level, this can generate a lot of data.

If the "Enable time" option is enabled, all lines in the logging will be preceded by the time of day.

16.3.1.6. Hide inactive

Checking this hides all steps that finished processing.

16.3.1.7. Safe mode

Places the transformation in Safe Mode. Additional row checking is enabled at runtime, see also Safe mode.

16.3.2. Job Buttons

16.3.2.1. Start job

This button begins execution of the current Job. Please note that Spoon launches attempts to launch the job from and XML-file or the Kettle repository. It is therefore necessary that the job is saved. The output of the execution is displayed in the Log Text part of the Log View.

16.3.2.2. Stop job

This button stops a running job.

16.3.2.3. Refresh log

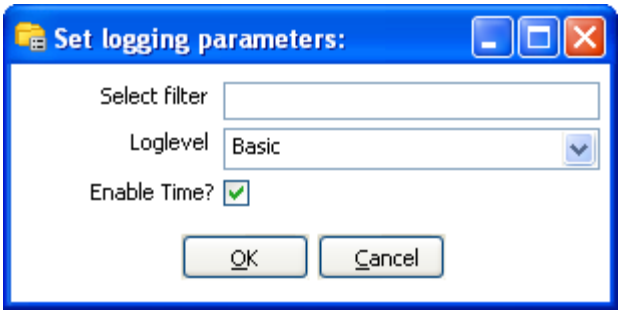
Refreshes the log window.

16.3.2.4. Clear log

This clears the text in the Log Text Window.

16.3.2.5. Log Settings

This is the "Log Settings" dialog:



If you put a text in the filter field, only the lines that contain this text will be shown in the Log Text window.

The "Log level" setting allows you to select the logging level. You can choose one of these:

- Error: Only show errors
- Nothing: Don't show any output
- Minimal: Only use minimal logging
- Basic: This is the default basic logging level
- Detailed: Give detailed logging output
- Debug: For debugging purposes, very detailed output.
- Row level: Logging at a row level, this can generate a lot of data.

If the "Enable time" option is enabled, all lines in the logging will be preceded by the time of day.

16.3.2.6. Auto-refresh

Enable this option to disable the logging window from updating all the time. You might want to do this when you're using a remote desktop (VNC, X11) over a slow network connection.

17. Grids

17.1. Description

Grids (tables) are used throughout the Spoon interface to enter or display information. This section describes common functions available when working with a Grid.

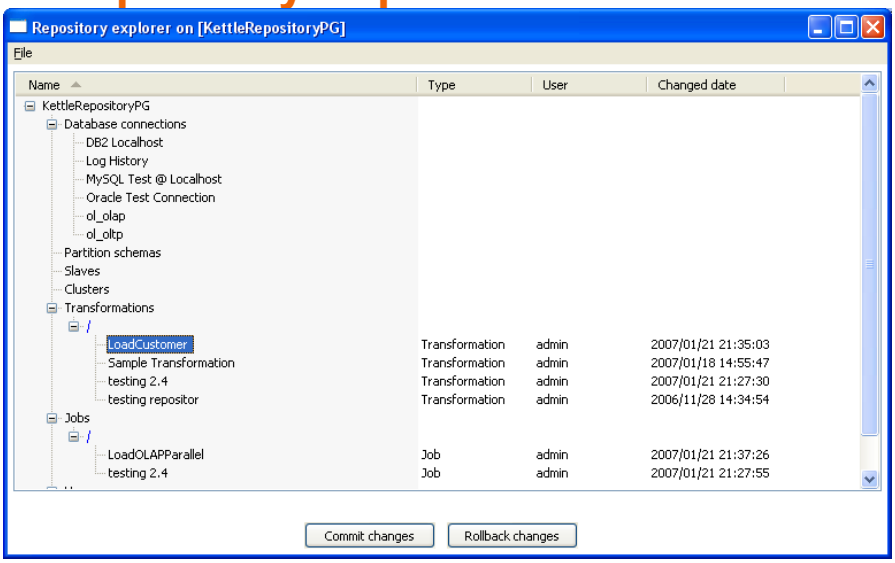
17.2. Usage

Click on a cell to begin editing that field. After pressing enter, you can navigate the grid by using the cursor keys. Pressing enter again allows you to edit the newly selected field.

The following table describes the functions available when you right-click on a cell in the grid:

Option	Description
Insert before this row	Inserts an empty row before the row you clicked on.
Insert after this row	*Inserts an empty row after the row you clicked on.*
Move the row up	Move the row you clicked on up. The keyboard shortcut for this is CTRL-UP
Move the row down	Move the row you clicked on down. The keyboard shortcut for this is CTRL-DOWN.
Optimal column size including header	Resize all columns so that it displays all values completely, including the header. The keyboard shortcut for this function is function key F3.
Optimal column size excluding header	Resize all columns so that it displays all values completely. The keyboard shortcut for this function is function key F4.
Clear all	Clears all information in the grid. You will be asked to confirm this operation.
Select all rows	Selects all rows in the grid. The keyboard shortcut for this function is CTRL-A.
Clear selections	Clears the selection of rows in the grid. The keyboard shortcut for this function is ESC.
Copy selected lines to clipboard	Copies the selected lines to the clipboard in a textual representation. These lines can then be exchanged with other programs such as spreadsheets or even other Spoon & Kettle dialogs. The keyboard shortcut for this function is CTRL-C.
Past clipboard to table	Insert the lines that are on the clipboard to the grid, right after the line on which you clicked. The keyboard shortcut for this function is CTRL-V.
Cut selected lines	Copies the selected lines to the clipboard in a textual representation. After that, the lines are deleted from the grid. The keyboard shortcut for this function is CTRL-X.
Delete selected lines	Deletes all selected lines from the grid. The keyboard shortcut for this function is DEL.
Keep only selected lines	If there are more lines to delete then there are to keep, simply select the lines you want to keep and use this function. They keyboard shortcut for this function is CTRL-K.
Copy field values to all rows	If all rows in the grid need to have the same value for a certain column, you can use this function to do this.
Undo	Undo the previous grid operation. The keyboard shortcut for this function is CTRL-Z.
Redo	Redo the next grid operation. The keyboard shortcut for this function is CTRL-Y.

18. Repository Explorer



18.1. Description

The repository Explorer shows you a tree view on the database repository to which you are connected. It allows you to examine and modify the content from the repository including:

- Connections
- Partition Schemas
- Slave servers
- Clusters
- Transformations

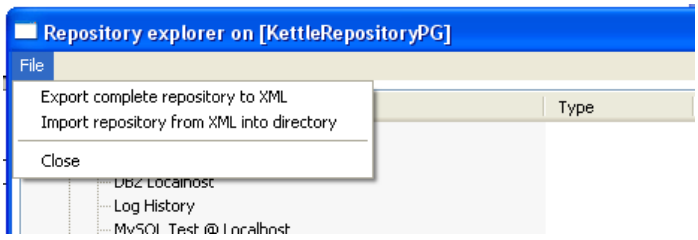
You can also click on the column header to sort content by name, object type, user or changed date.

18.2. Right click functions

Right clicking on an object in the repository will bring up basic functions such as open, delete and rename objects.

18.3. Backup / Recovery

It is possible to export the complete repository in XML: See the options available in the file menu of the repository explorer.



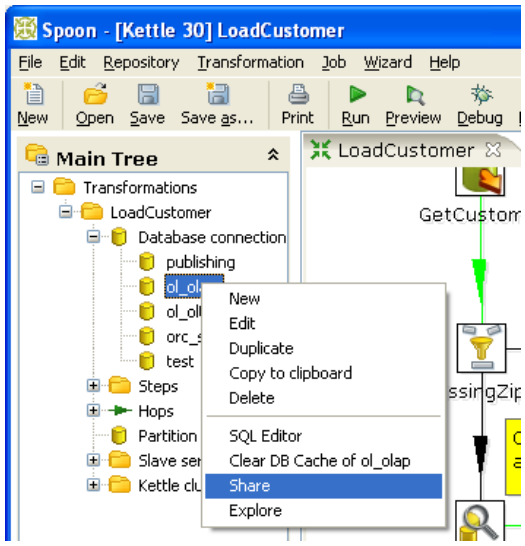
NOTE: *you can restore the objects from a backed up repository anywhere in the target repository directory tree.*

19. Shared objects

A variety of objects can now be placed in a shared objects file on the local machine. The default location for the shared objects file is $HOME/.kettle/shared.xml. Objects that can be shared using this method include:

- Database connections
- Steps
- Slave servers
- Partition schemas
- Cluster schemas

To share one of these objects, simply right-click on the object in the tree control on the left and choose "share".



Note: *The location of the shared objects file is configurable on the "Miscellaneous" tab of the Transformation|Settings dialog.*

20. APPENDIX A: LGPL License

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge
for this service if you want); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes

to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We want to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by

limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a=
"work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for afee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of

this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.
In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of ordinary GNU General Public License has appeared, then you can specify that version instead if you want.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy. This option is useful when you wish to copy part of the code of= the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is

therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one of these things:

> a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that

system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you want to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our
decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> <one line to give the library's name and a brief idea of what it does.>
> Copyright (C) <year> <name of author>
>
> This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public
> License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.
>
> This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.
>
> You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
        Yoyodyne, Inc., hereby disclaims all copyright interest in the library
        `Frob' (a library for tweaking knobs) written by James Random Hacker.

        <signature of Ty Coon>, 1 April 1990
        Ty Coon, President of Vice
```

That's all there is to it!