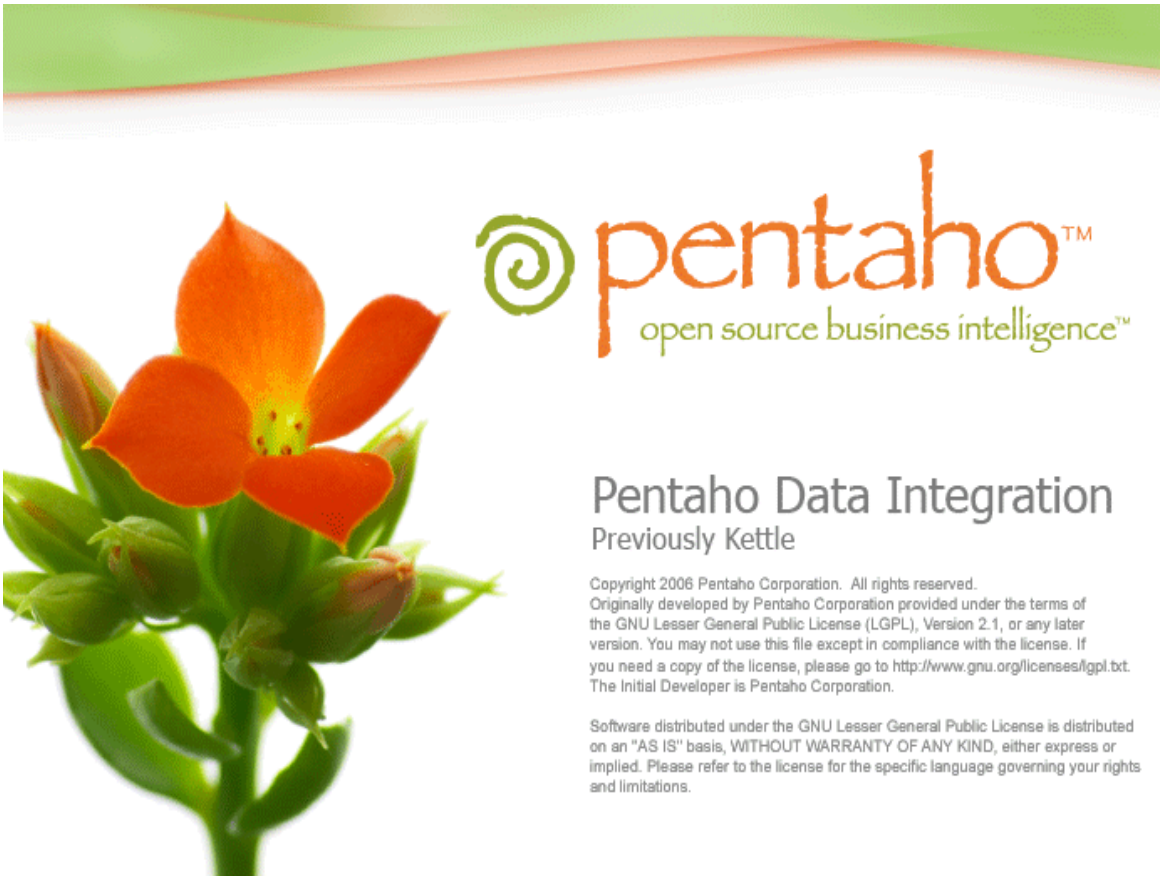


Chef version 2.3.0

http://www.pentaho.org

Table of contents

1. CHEF

1.1. What is Chef?

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.

Chef is a graphical user interface for that allows you to design jobs that can be executed with the Kettle tool Kitchen. Kitchen is a job execution engine that is capable of performing a multitude of functions such as: execute transformations, execute jobs, verify file existence, get files using FTP, SFTP, HTTP, ...

NOTE: *For a complete description of Kitchen, please check out the Kitchen documentation.*

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

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

1.2. 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-2.3.0.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
```

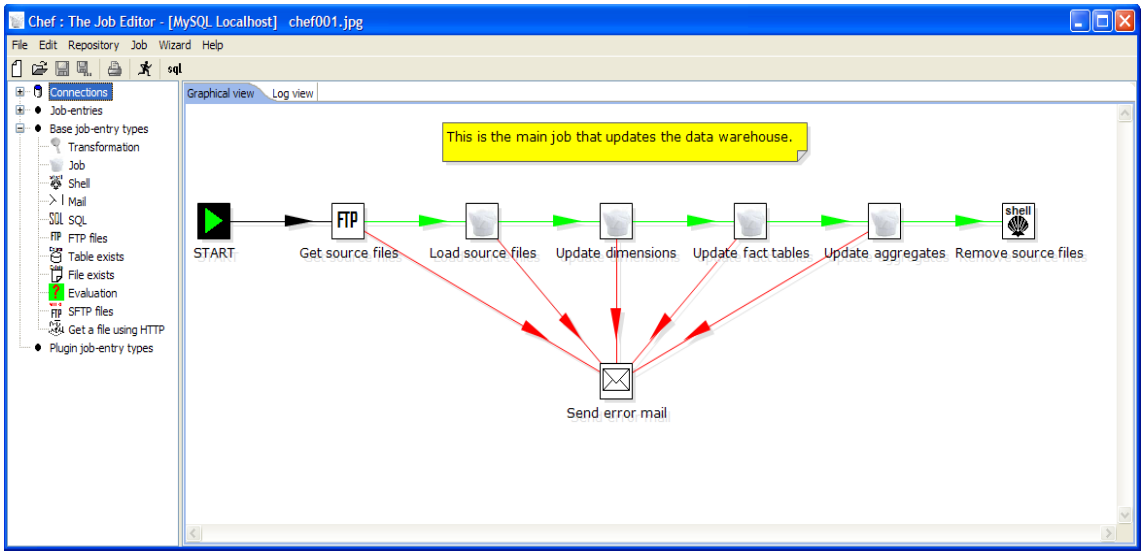
1.3. Launching Chef

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

- ✓ Chef.bat: launch Chef on the Windows platform.
- ✓ chef.sh: launch Chef on a Unix-like platform: Linux, Solaris, AIX, HP-UX, OS X

If you want to make a shortcut under the Windows platform an Icon is provided: “chef.ico” to set the correct icon. Simply point the shortcut to the Chef.bat file.

1.4. Screen shot



The following items are visible in Chef: Connections, Job-entries, job-entry types, Graphical View and the Log View.
These items are described in detailed in the chapters below: 2. Database Connections, 7. Job Entries, 8. Graphical View and 9. Log View.

1.5. Command line options

IMPORTANT NOTES:

- *On Windows system, the use of the minus ("-") in the options causes problems as well as the equal sign ("="). Because of this, from version 2.2.2 on, you can also use this format or any combination of /,- and :,= for example /option:value*
- *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.*

These are the command line options that you can use.

```
-file=filename
```

This option runs the job defined in the XML file. (.kjb : Kettle Job)

```
-log=Logging Filename
```

Specifies the log file. The default is the standard output.

```
-rep=Repository name
```

Connect to the repository with name "*Repository name*".
You also need to specify the options –user, –pass and –trans.
The repository details are loaded from the file repositories.xml in the local directory or in the Kettle directory: <homedirectory>/.kettle/
You can also specify this option in the form of environment variable KETTLE_REPOSITORY. This will allow auto-login to the repository of your choice.

```
-user=Username
```

This is the username with which you want to connect to the repository.
You can also specify this option in the form of environment variable KETTLE_USER. This will allow auto-login to the repository of your choice.

```
-pass=Password
```

The password to use to connect to the repository
You can also specify this option in the form of environment variable KETTLE_PASSWORD. This will allow auto-login to the repository of your choice.

```
-job=Job Name
```

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

NOTE: *It's important that if spaces are present in the option values, you use quotes or double quotes to keep them together. This behavior is OS and Shell dependent.*

1.6. Repository

A Kettle repository can contain among other things transformations and jobs. This means that in order to load a job 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 Chef.

The information concerning repositories in stored in file called “repositories.xml”. This file resides in the hidden directory “.kettle” in your default home directory.

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

IMPORTANT: *The default password for the admin user is also admin. You should change this default password right after the creation of the repository. You can change the password using “Edit current user” or with the Repository Explorer (see 11. Repository Explorer)*

1.7. License

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

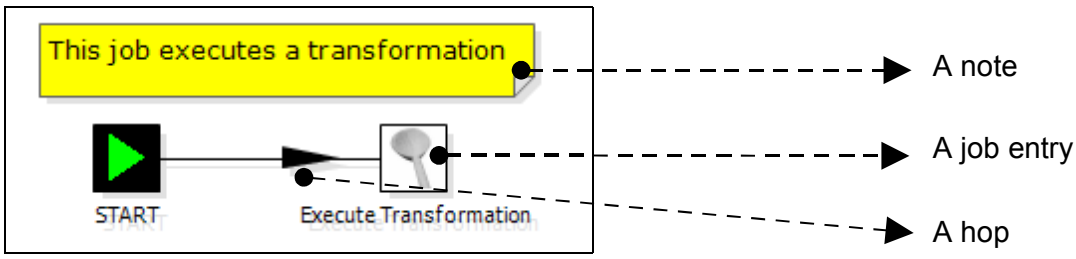
```
Copyright (C) 2006 Pentaho Corporation

    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
```

1.8. Notes

You can put notes on the graphical view everywhere simply by clicking right on the canvas and selecting “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.
Removing a note can be done by a right click on the note and by selecting “Delete note”.

1.9. Definitions

- ✓ Job Entry: A job entry is one part of a job and performs a certain
- ✓ 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 piece of information that can be added to a job



1.10. Toolbar

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

Icon	*Meaning*
	New job
	Open job from file if you're not connected to a repository or from the repository if you are connected to one.
	Save the job to a file or to the repository.
	Save the job under a different name or filename.
	Print: you will be presented with a print-dialog asking you to specify the number of pages, margins etc.
	Run the job: runs the current job from XML file or repository.

1.11. Options

Please see the options described in Spoon-2.3 - 2.14 Options

2. DATABASE CONNECTIONS

2.1. Description

A connection describes the method by which Kettle can connect to a database. The top entries in the tree on the left describe the available connections.

For a complete description on how to use database connection, please see Spoon-2.3.pdf, chapter 3. Database Connections

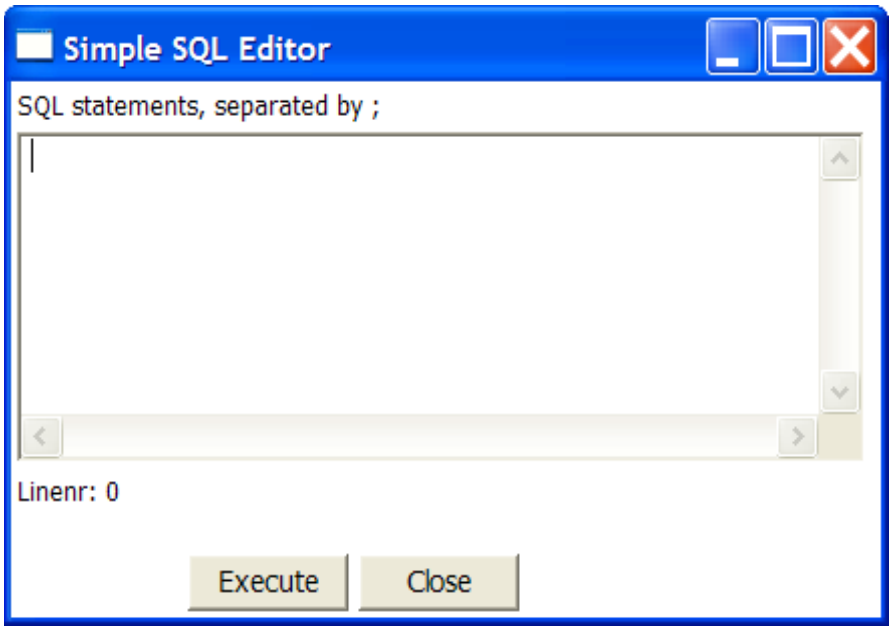
3. SQL EDITOR

3.1. Description

Sometimes a simple SQL Editor can be nice to have. Especially when you're creating tables, dropping indexes and modifying fields. The simple SQL editor supplied in Chef, allows you to do this. In fact, most of the DDL (Data Definition Language) such as “create/alter table”, “create index” and “create sequence” SQL commands are created automatically for you via the SQL Editor window.

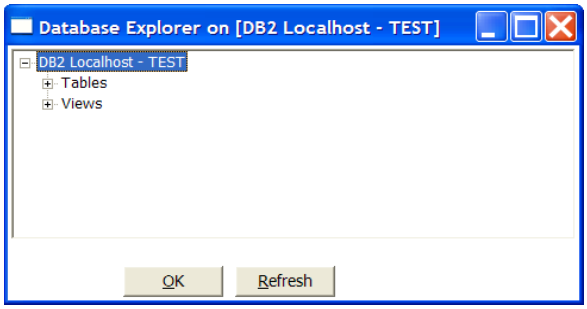
NOTE: *Multiple SQL Statements have to be separated by semi-colons (;).*
NOTE: *Before these SQL Statements are sent to the database to be executed, Chef removes returns, line-feeds and the separating semi-colons.*
NOTE: *Kettle clears the database cache for the database connection on which you launch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

3.2. Screen shot



4. DATABASE EXPLORER

4.1. Screenshot



4.2. Description

The database explorer allows you to explore the database to which the database connection points. At the moment, it only shows available tables and the catalog and/or schema to which the table belongs.
It is possible to click right on a shown table or view (lowest level in the tree) and select one of the following options:

- ✓ Display the first 100 rows of the table (also available through double-click on table name)
- ✓ Display the first … lines of the table
- ✓ Show the size (in rows) of the table.
- ✓ Show layout of the table
- ✓ Generate the DDL statement (create table ...) for this table.
- ✓ Generate the DDL statement (create table ...) for this table on another database connection
- ✓ Show the SQL statement to read from this table. (in SQL Editor)

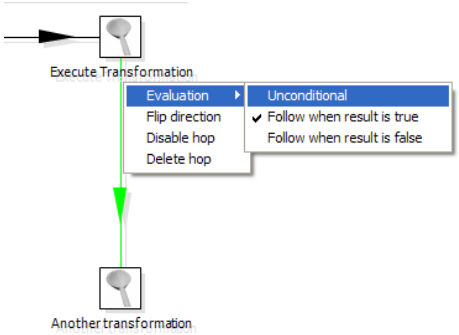
5. JOB HOPS

5.1. Description

A job hop connects one job entry with another. The execution order is indicated with an arrow on the graphical view pane. A job hop can be enabled or disabled. (For testing purposes for example).
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 false, ...
- "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, ...

5.2. Screenshot



5.3. Creating A Hop

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

- ✓ Dragging on the Graphical View between 2 job entries while using the middle mouse button.
- ✓ Dragging on the Graphical View between 2 job entries while pressing the SHIFT key and using the left mouse button.
- ✓ Selecting 2 job entries in the graphical view (CTRL + left mouse click), clicking right on a job entry and selecting "new hop"

5.4. Splitting A Hop

You can easily insert a new job entry into a hop between 2 job entries by dragging the job entry (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 job entries that have not yet been connected other job entries.

5.5. Loops

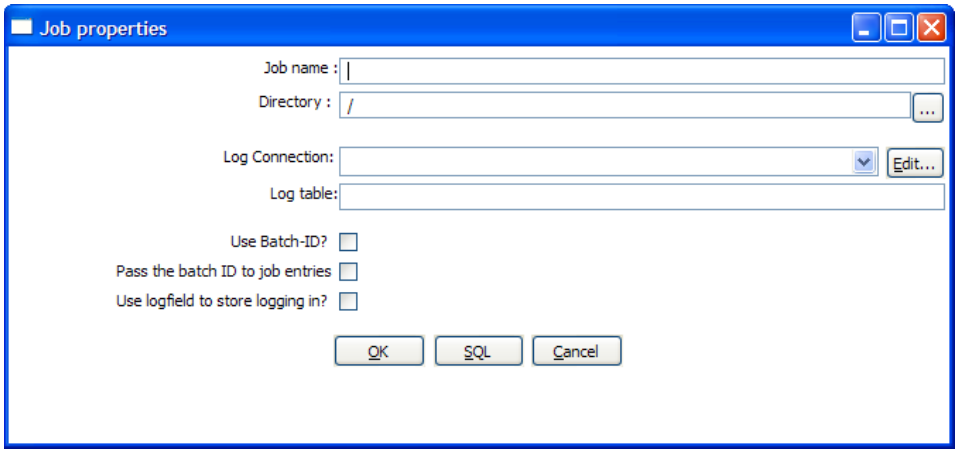
Loops **are** allowed in jobs because Chef executes job entries sequentially. Just make sure you don't build endless loops. Please check out Evaluation job entry (7.2.10. Evaluation). This job entry can help you exit closed loops based on the number of times a job entry was executed.

6. JOB SETTINGS

6.1. Description

There are a few options that control how a job is behaving and how it is logging what it is doing.

6.2. Screenshot



6.3. Options

Option	*Description*
job name	the name of the job. This is required information if you want to save to a repository.
Directory	the directory in the repository in which you want to save the job.
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)

6.4. Extra

✓ SQL button: generates the SQL needed to create the logging table and allows you to execute this SQL statement.

7. JOB ENTRIES

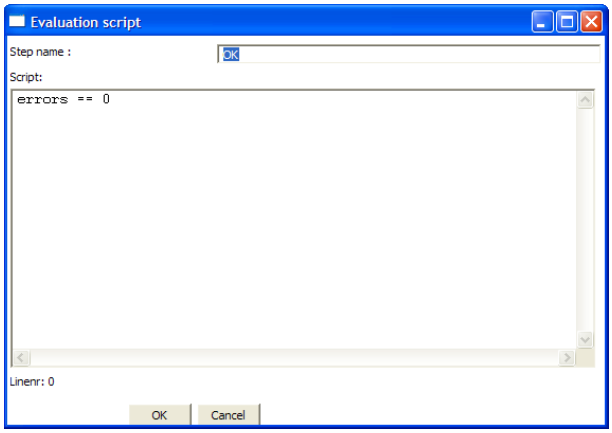
7.1. Description

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.

7.2. Job Entry Types

7.2.1. Special Job Entries

7.2.1.1. Screenshot

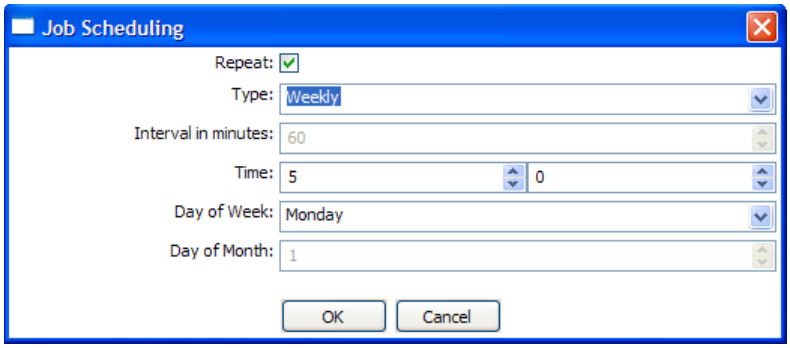


7.2.1.2. Icons

7.2.1.3. General description

7.2.1.3.1. Start

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. If you edit this job entry you'll get this screen:



7.2.1.3.2. Dummy

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.

7.2.1.3.3. OK

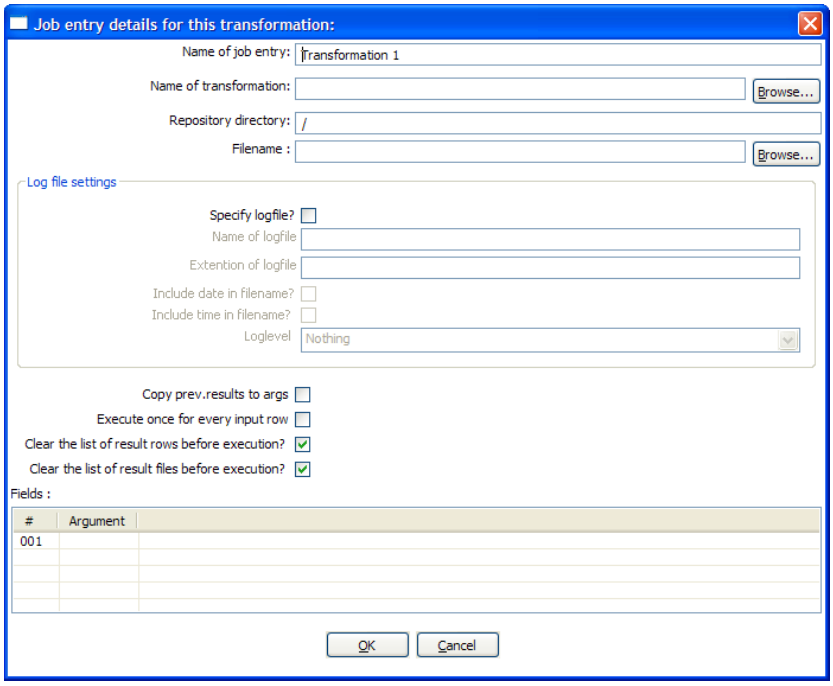
Checks if the number of errors from a previous job entry was 0 and results to true if this is the case.
See also the “Evaluation” job entry.

7.2.1.3.4. ERROR

Checks if the number of errors from a previous job entry was different from 0 and results to true if this is the case. See also the “Evaluation” job entry.

7.2.2. Transformation

7.2.2.1. Screenshot



7.2.2.2. Icon

7.2.2.3. General description

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

7.2.2.4. Options

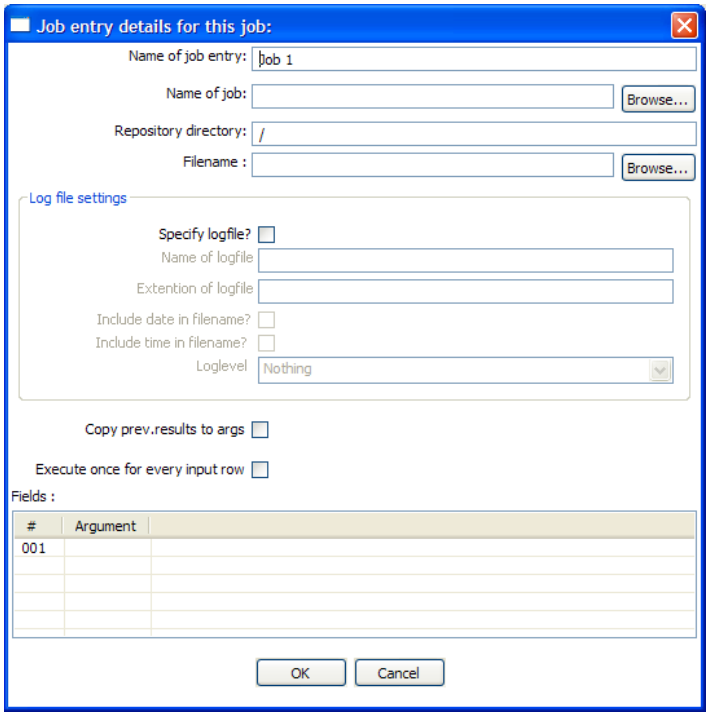
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Name of transformation	The name of the transformation to start.
Repository directory	The 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)

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 transformation. See also the logging window in 9.4.4. Log Settings
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.
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.*

7.2.3. Job

7.2.3.1. Screenshot



7.2.3.2. Icon

7.2.3.3. 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.*

7.2.3.4. Options

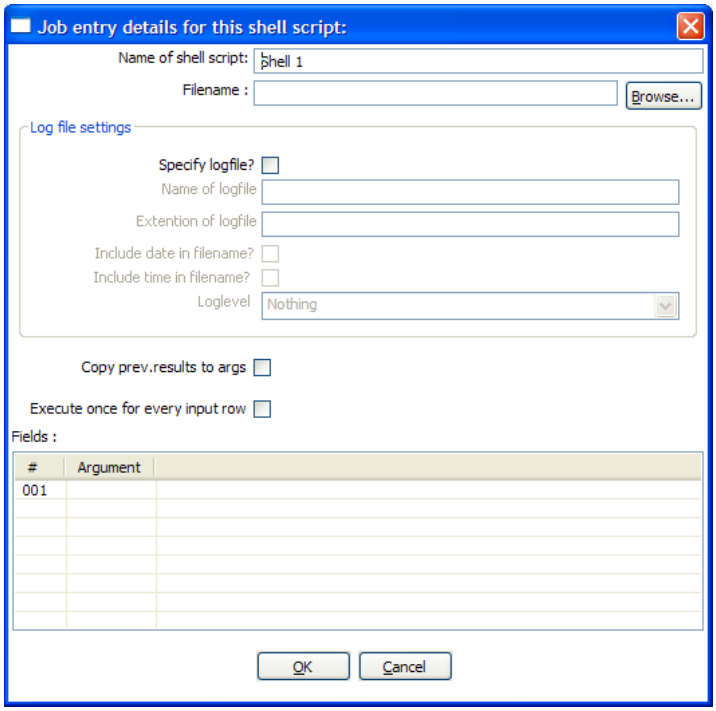
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Name of transformation	The name of the job to start.
Repository directory	The directory in 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)

Option	*Description*
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 job. See also the logging window in 9.4.4. Log Settings
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.*

7.2.4. Shell

7.2.4.1. Screenshot



7.2.4.2. Icon

7.2.4.3. 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 (since release 2.3.0) 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 now preceded by "CMD.EXE /C" (NT/XP/2000) or "COMMAND.COM /C" (95,98).*

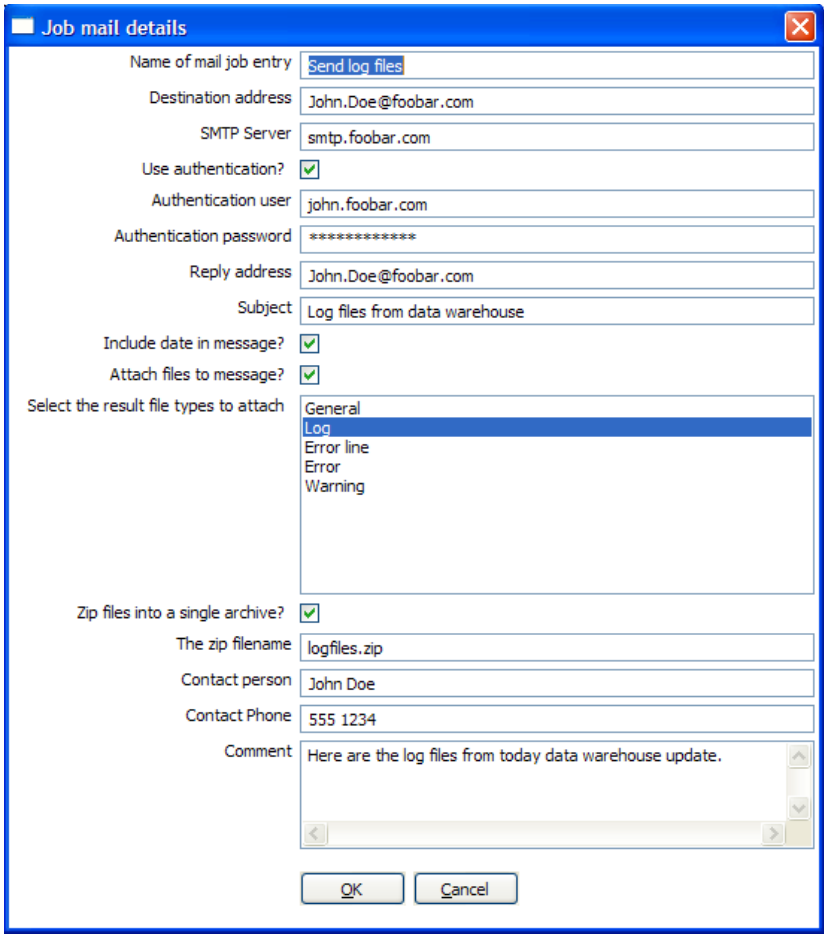
7.2.4.4. Options

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 filename of the shell script to start.
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)
Logging level	Specifies the logging level for the execution of the shell. See also the logging window in 9.4.4. Log Settings
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)
Arguments	Specify the strings to use as arguments for the shell script.
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)

7.2.5. Mail

7.2.5.1. Screenshot



7.2.5.2. Icon

7.2.5.3. General description

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

7.2.5.4. Options

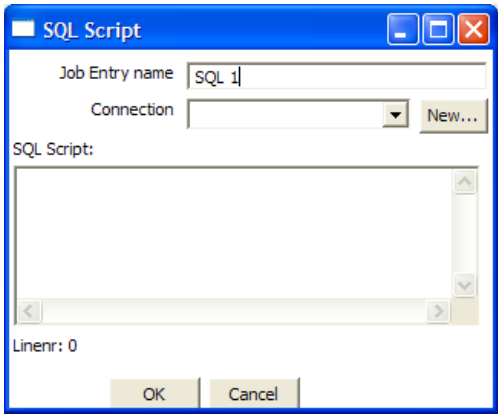
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Destination address	The destination for the e-Mail
Use authentication	Check this if your SMTP server requires you to authenticate yourself. Please note that secure authentication **is not yet supported** in version 2.3.0.
Authentication user	The user name to authenticate with

Option	***Description***
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: A*ll text fields can be specified using (environment and Kettle) variables, possibly set in a previous transformation using the Set Variable step.*

7.2.6. SQL

7.2.6.1. Screenshot



7.2.6.2. Icon

7.2.6.3. General description

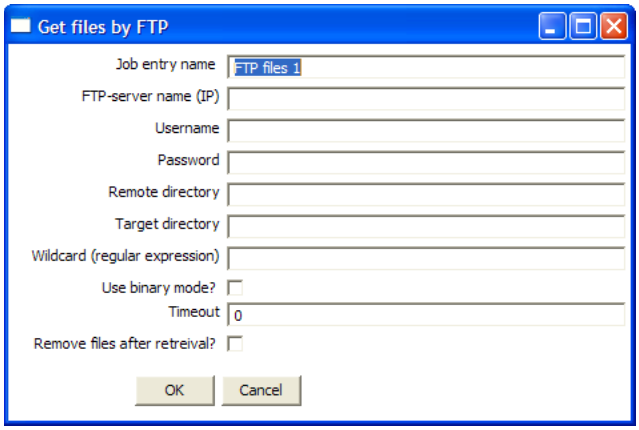
You can use the *SQL* job entry to execute an SQL script.
This means a number of SQL statements separated by ;

7.2.6.4. 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.
Connection	The database connection to use
SQL script	The SQL script to execute

7.2.7. FTP

7.2.7.1. Screenshot



7.2.7.2. Icon

7.2.7.3. General description

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

7.2.7.4. Options

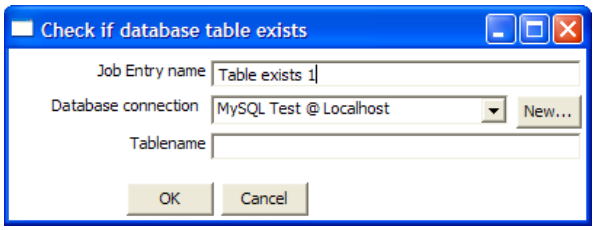
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
FTP server name	The name of the server or the IP address
User name	The user name to log into 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.

7.2.8. Table Exists

7.2.8.1. Screenshot



7.2.8.2. Icon

7.2.8.3. General description

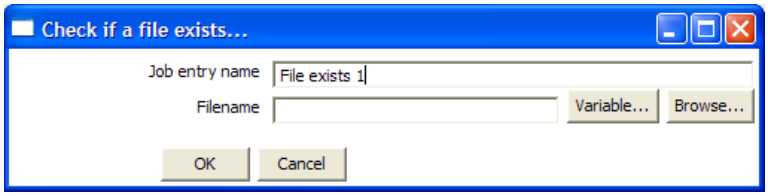
You can use the *Table Exists* job entry to verify if a certain table exists on a database.

7.2.8.4. 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

7.2.9. File Exists

7.2.9.1. Screenshot



7.2.9.2. Icon

7.2.9.3. General description

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

7.2.9.4. Options

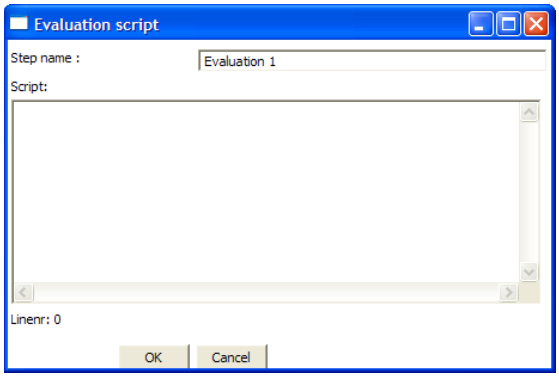
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
Filename	The name and path of the file to check for

7.2.9.5. Extra

- Variable: select the variable to use as filename
- Browse: look for the file on the file system

7.2.10. Evaluation

7.2.10.1. Screenshot



7.2.10.2. Icon

7.2.10.3. General description

You can use the *Evaluation* job entry to calculate a boolean variable. This variable can then be used to determine which next step will be executed.
The following variables are available to the user:

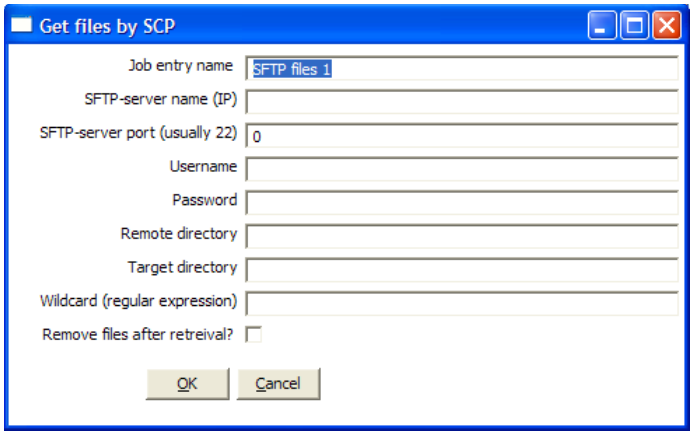
- errors : number of errors in the previous job entry (long)
- lines_input : number of rows read from database or file (long)
- lines_output : number of rows written to database or file (long)
- lines_updated : number of rows updated in a database table (long)
- lines_read : number of rows read from a previous transformation step (long)
- lines_written : number of rows written to a next transformation step (long)
- files_retrieved : number of files retrieved from an FTP server (long)
- exit_status : the exit status of a shell script (integer)
- nr (integer) : the job entry number. Increments at every next job entry.
- is_windows : true if Kettle runs on MS Windows (boolean)

7.2.10.4. 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.
Script	The Javascript that needs to be run to evaluate. The last statement in the script needs to result in a boolean

7.2.11. SFTP

7.2.11.1. Screenshot



7.2.11.2. Icon

7.2.11.3. General description

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

7.2.11.4. Options

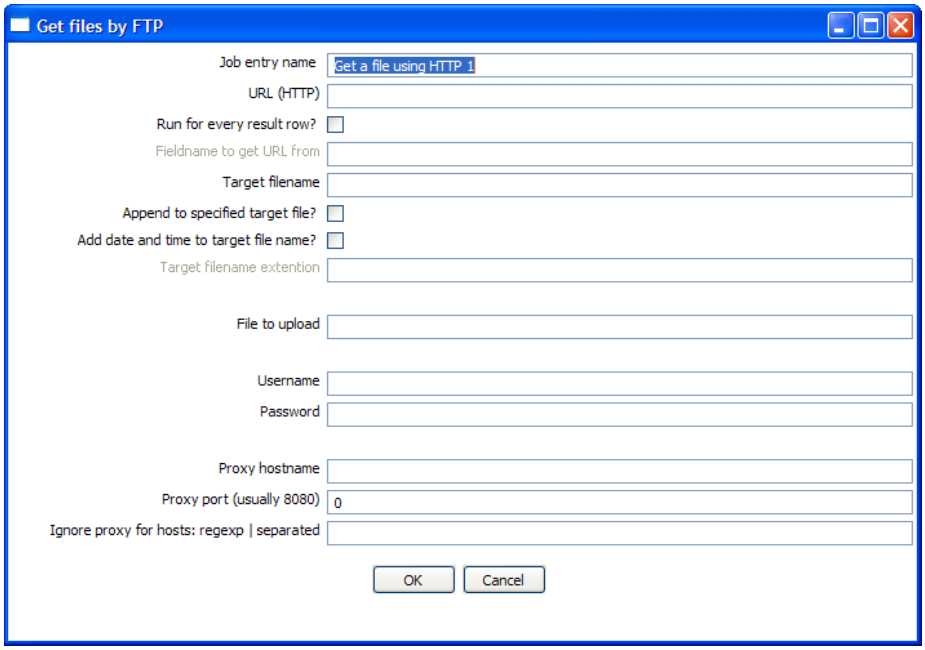
Option	***Description***
Name of the job entry	The name of the job entry. This name has to be unique in a single job. A job entry can be placed several times on the canvas, however it will be the same job entry.
SFTP-server name (IP)	The name of the SFTP server or the IP address
SFTP port	The TCP port 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 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 on the SFTP server, but only after all selected files have been successfully transferred.

7.2.12. HTTP

7.2.12.1. Screenshot



7.2.12.2. Icon

7.2.12.3. General description

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

7.2.12.4. 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 (HTTP)	The URL to use (for example: http://www.kettle.be/index.html)
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”
Fieldname to get URL from	The fieldname in the result rows to get the URL from
Target filename	The target filename
Append to 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 extention	Specify the target filename extension in case you're adding a date and time to the filename

Option	***Description***
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	The HTTP proxy server name or IP address
Proxy port	The HTTP proxy port to use (usually 8080)
Ignore proxy for hosts	Specify a regular expression matching the hosts you want to ignore, | separated. For example 127\.0\..*

8. GRAPHICAL VIEW

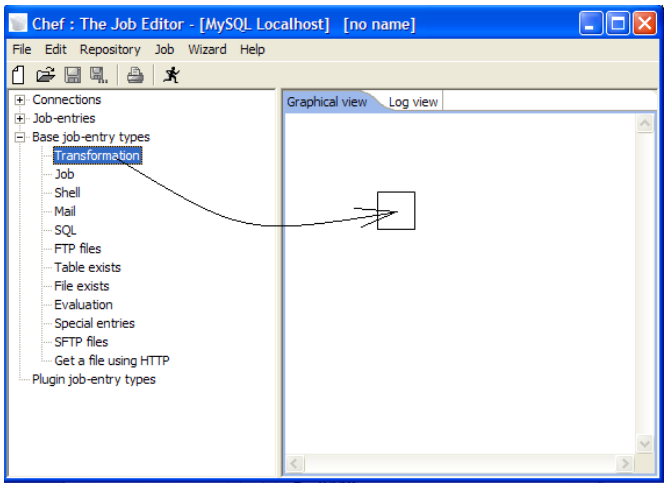
8.1. Description

The Graphical View is the canvas on which jobs are drawn.
It shows an easy to understand representation of the work that needs to be done and the flow of the data.

8.2. Adding job entries

8.2.1. Dragging

Adding steps to a transformation 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 job entries when you let go of the button.
When you let go of the mouse button the selected job entry (Transformation) will become part of the job.

8.3. Hiding and deleting a job entry

A single job entry can be placed multiple times on the canvas.
A job entry can be removed from the job (hidden), but can only be deleted when all copies have been hidden. Click right on a job entry on the canvas to hide it.

8.4. Job entry options (right click popup menu)

8.4.1. Launch Spoon/Chef

This opens a Spoon or Chef window displaying the selected transformation or job.

8.4.2. Edit job entry

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

8.4.3. Edit job entry description

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

8.4.4. Duplicate job entry

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

8.4.5. Delete step

This will permanently remove the step from the transformation.

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

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

8.4.7. align / distribute

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

8.4.8. Detach entry

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

8.4.9. Delete all copies of this entry.

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

8.5. Adding hops

On the graphical view the quickest way to create a new hop is by dragging with the mouse from one job entry to another using the middle button.
You can also drag the left button and press the SHIFT key at the same time.

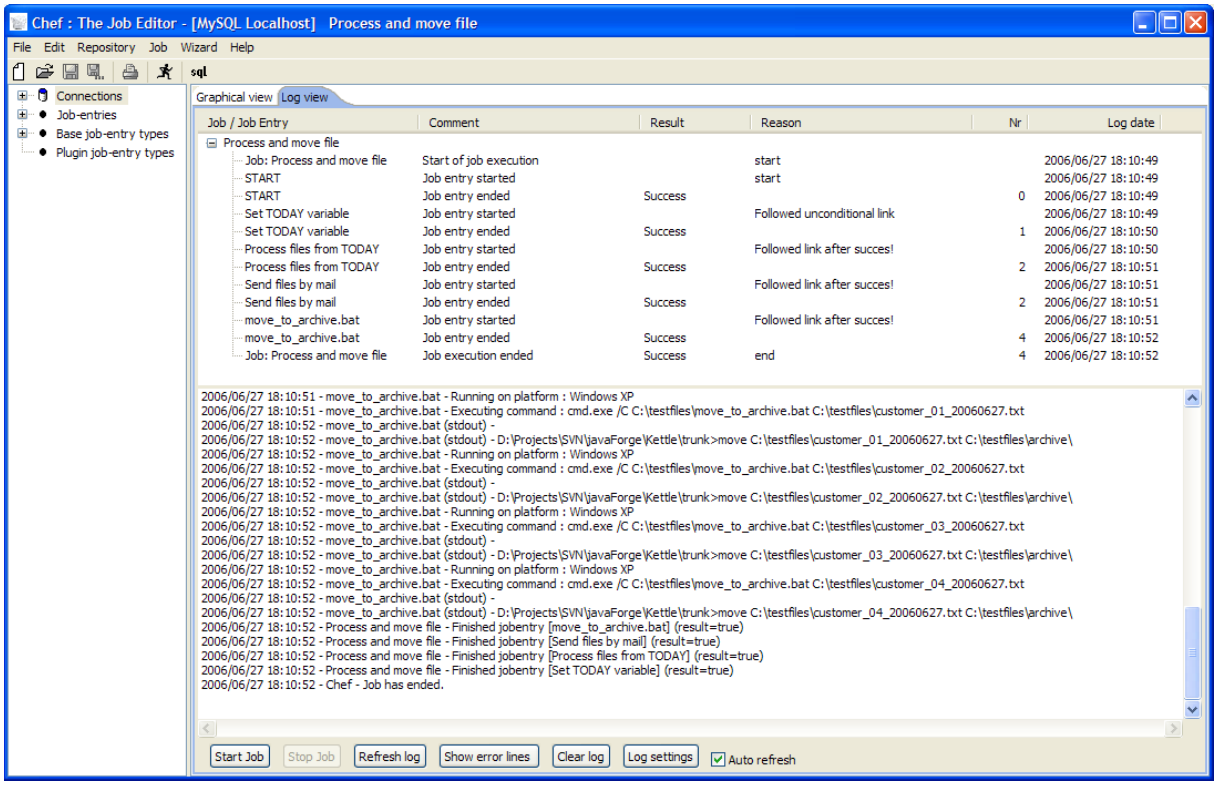
For a more complete explanation regarding hops, please check out 5. Job Hops

9. LOG VIEW

9.1. Description

The Log View shows what's happening when a job is running.
First of all it shows all the details of the completed job entries. Secondly, it shows the log as it would be shown if the job would be launched by Kitchen.

9.2. Screenshot



9.3. Log Grid

The log grid is actually a tree that offers a hierarchical view on the execution of a job. The following items are shown:

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

9.4. Buttons

9.4.1. Start job

This button starts the job you're designing. Please note that Chef tries to launch this as Kitchen would: from XML-file or 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.

9.4.2. Stop job

This button stops a running job.

9.4.3. Refresh log

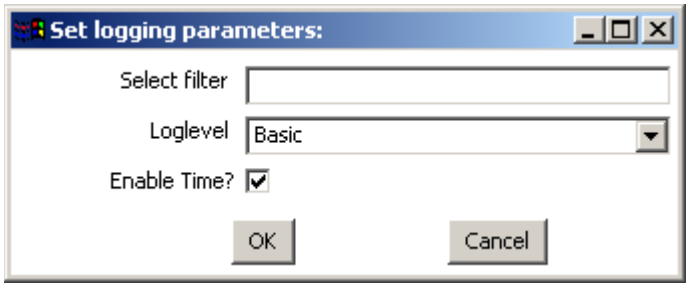
Refreshes the log window.

9.4.4. Clear log

This clears the text in the Log Text Window.

9.4.5. 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.

9.4.6. Auto-refresh

If you rather not have the logging window updating all the time you can disable this option. You might want to do this when you're using a remote desktop (VNC, X11) over a slow network connection.

10. GRIDS

10.1. Description

Grids are used everywhere in Chef & Kettle. They are used to enter or display information.

For a more complete description of grids, please see Spoon-2.3.pdf

11. REPOSITORY EXPLORER

11.1. 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.

For a more complete description of the repository explorer, please see Spoon-2.3.pdf

12. APENDIX A

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 wish); 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 wish 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 a
fee.

  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 the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) 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 wish 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!
```