Pentaho Data Integration
Previously Kettle

Pentaho Data Integration
version 2.3.0

A new batch of features!

Key differences with Kettle 2.2.2

List of changes on 15-06-'06

Compiled by Matt Casters, mcasters@pentaho.org

Send additional changes you found to this address.

Index

1. Changes summary

1.1. Preface

Over the last 5 months, a lot of things have happened to the Pentaho Data Integration project. Numerous developers joined the project and there were bug fixes provided by people in various regions of the world. In fact there have been so many changes to the code-base that it is a challenge to write this very summary. We believe one of the reasons for the many changes is the popularity that "Kettle" gained in this period. This has led to a long list of challenges that the software received and met in the course of the 2.3.0 development.

1.2. Overview

This are the most notable changes that have been made:

- Overall performance enhancements
- New error handling code (with replay) for reading text files and excel documents
- New steps: Delete, Value mapper, Set Variable, Get Variable, Get File names, Get files from result, Set files in result and the Blocking step.
- Search meta-data functionality in Spoon: look for a value or parameter.
- All sorts of parameterizing functionality in transformations and jobs
- The possibility to export and import of repositories, even in batch for backup.
- Localization of the code into Chinese, French, German and Dutch
- Many enhancements to the existing steps and job entries.
- Improved logging capabilities
- Easier fixed target table mapping
- Improved look & feel for all platforms
- ...

2. General changes

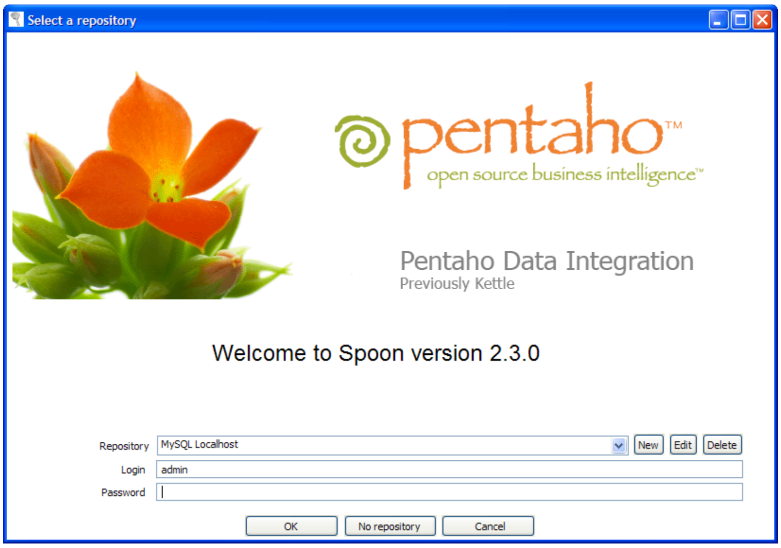
2.1. New splash screen

Pentaho Data Integration
Previously Kettle

Copyright 2006 Pentaho Corporation. All rights reserved.
Originally developed by Pentaho Corporation provided under the terms of the GNU Lesser General Public License (LGPL), Version 2.1, or any later version. You may not use this file except in compliance with the license. If you need a copy of the license, please go to http://www.gnu.org/licenses/lgpl.txt.
The Initial Developer is Pentaho Corporation.

Software distributed under the GNU Lesser General Public License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Please refer to the license for the specific language governing your rights and limitations.

2.2. New login screen



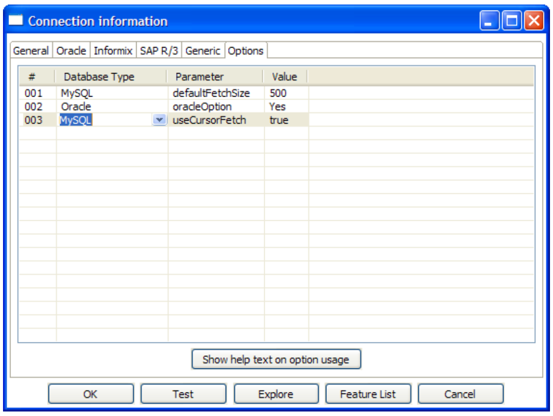
Pentaho Data Integration
Previously Kettle

Welcome to Spoon version 2.3.0

Repository: MySQL Localhost
Login: admin
Password:

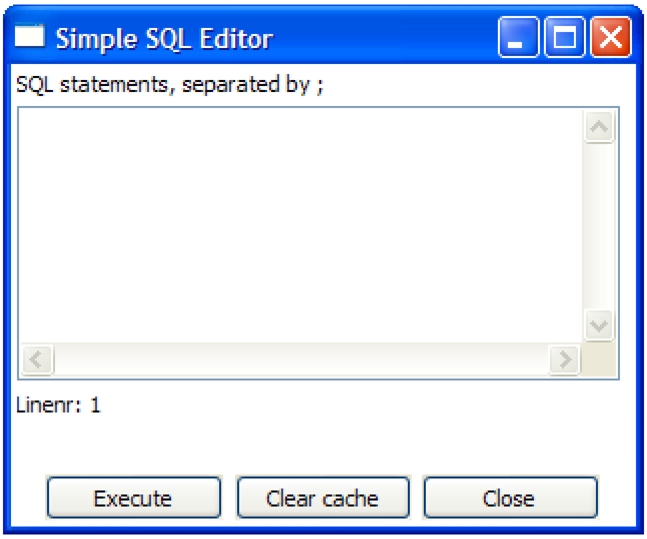
2.3. Databases

- Added support for ExtenDB (http://www.extendb.com/, patch by Mason Sharp <msharp@extendb.com>)
- Extra database specific options can be specified. These options are added to the connection settings.(through the URL or by setting Properties at connect time)



#	Database Type	Parameter	Value
001	MySQL	defaultFetchSize	500
002	Oracle	oracleOption	Yes
003	MySQL	useCursorFetch	true

- Better support for handling reserved keyword quoting, as well as quoting of tables and fields with spaces, slashes, dashes, plus, etc. in it.
- Repository support on most (if not all) supported database platforms.
- Better handling of schema and catalog names in the repository explorer.
- We also added a "clear cache" button to the SQL Editor because the database cache often is no longer valid when working with 3rd party tools to create or alter tables. Because this is where the problem becomes visible, this is where we added the button:



2.4. Data grids

- Serious performance upgrades for large grids
- Selected lines are now bright blue, no longer gray.

- Clicking below the last line automatically adds a new line
- Going beyond the last line with the cursor automatically adds a new line.

2.5. Platform, Look & feel

- Both Linux and OS-X are now brought alongside of the Windows platform in terms of functionality and looks.
- Most (if not all) OS-X and Linux specific bugs are gone.
- Thanks to the new SWT 3.2 libraries, the look and feel of Windows is taken over automatically, giving a nicer user-experience on that platform. These are to be seen in the screen shots in this document.

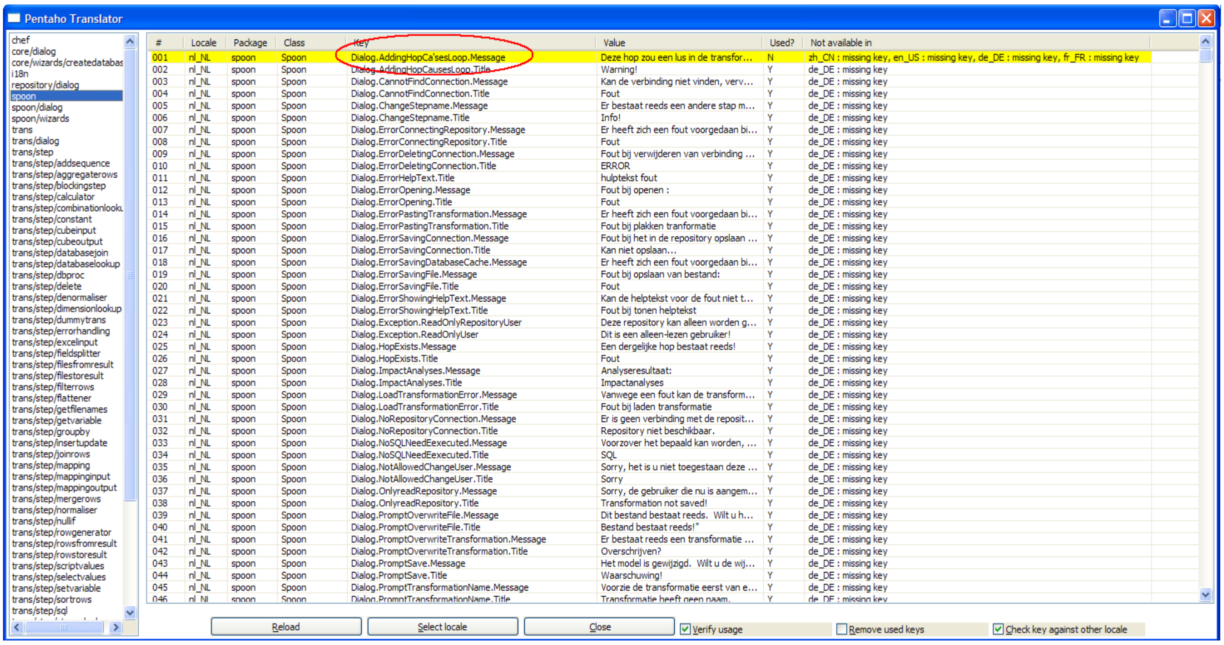
2.6. Localizations

Because of popular demand, we localized version 2.3.0 to allow us to translate the externalized strings. So far there are 4 languages we have available or partly available:

- English (100%)
- Simplified Chinese (>90%)
- French (>75%)
- German (>40%)
- Dutch (10%)

The next languages that will become available are most likely Spanish and Portuguese.

To keep an overview of the translation efforts, we have created a new program called Pentaho Translator that allows you to browse the keys in the different languages and source packages. It makes it easier to detect typos, unused keys, etc.:



#	Locale	Package	Class	Key	Value	Used?	Not available in
001	nl_NL	spoon	Spoon	Dialog.AddingHopCa'sesLoop.Message	Deze hop zou een lus in de transfor…	N	zh_CN : missing key, en_US : missing key, de_DE : missing key, fr_FR : missing key
002	nl_NL	spoon	Spoon	Dialog.AddingHopCausesLoop.Title	Warning!	Y	de_DE : missing key
003	nl_NL	spoon	Spoon	Dialog.CannotFindConnection.Message	Kan de verbinding niet vinden, verv…	Y	de_DE : missing key
004	nl_NL	spoon	Spoon	Dialog.CannotFindConnection.Title	Fout	Y	de_DE : missing key
005	nl_NL	spoon	Spoon	Dialog.ChangeStepname.Message	Er bestaat reeds een andere stap m…	Y	de_DE : missing key
006	nl_NL	spoon	Spoon	Dialog.ChangeStepname.Title	Info!	Y	de_DE : missing key
007	nl_NL	spoon	Spoon	Dialog.ErrorConnectingRepository.Message	Er heeft zich een fout voorgedaan bi…	Y	de_DE : missing key
008	nl_NL	spoon	Spoon	Dialog.ErrorConnectingRepository.Title	Fout	Y	de_DE : missing key
009	nl_NL	spoon	Spoon	Dialog.ErrorDeletingConnection.Message	Fout bij verwijderen van verbinding …	Y	de_DE : missing key
010	nl_NL	spoon	Spoon	Dialog.ErrorDeletingConnection.Title	ERROR	Y	de_DE : missing key
011	nl_NL	spoon	Spoon	Dialog.ErrorHelpText.Title	hulptekst fout	Y	de_DE : missing key
012	nl_NL	spoon	Spoon	Dialog.ErrorOpening.Message	Fout bij openen :	Y	de_DE : missing key
013	nl_NL	spoon	Spoon	Dialog.ErrorOpening.Title	Fout	Y	de_DE : missing key
014	nl_NL	spoon	Spoon	Dialog.ErrorPastingTransformation.Message	Er heeft zich een fout voorgedaan bi…	Y	de_DE : missing key
015	nl_NL	spoon	Spoon	Dialog.ErrorPastingTransformation.Title	Fout bij plakken tranformatie	Y	de_DE : missing key
016	nl_NL	spoon	Spoon	Dialog.ErrorSavingConnection.Message	Fout bij het in de repository opslaan …	Y	de_DE : missing key
017	nl_NL	spoon	Spoon	Dialog.ErrorSavingConnection.Title	Kan niet opslaan…	Y	de_DE : missing key
018	nl_NL	spoon	Spoon	Dialog.ErrorSavingDatabaseCache.Message	Er heeft zich een fout voorgedaan bi…	Y	de_DE : missing key
019	nl_NL	spoon	Spoon	Dialog.ErrorSavingFile.Message	Fout bij opslaan van bestand:	Y	de_DE : missing key
020	nl_NL	spoon	Spoon	Dialog.ErrorSavingFile.Title	Fout	Y	de_DE : missing key
021	nl_NL	spoon	Spoon	Dialog.ErrorShowingHelpText.Message	Kan de helptekst voor de fout niet t…	Y	de_DE : missing key
022	nl_NL	spoon	Spoon	Dialog.ErrorShowingHelpText.Title	Fout bij tonen helptekst	Y	de_DE : missing key
023	nl_NL	spoon	Spoon	Dialog.Exception.ReadOnlyRepositoryUser	Deze repository kan alleen worden g…	Y	de_DE : missing key
024	nl_NL	spoon	Spoon	Dialog.Exception.ReadOnlyUser	Dit is een alleen-lezen gebruiker!	Y	de_DE : missing key
025	nl_NL	spoon	Spoon	Dialog.HopExists.Message	Een dergelijke hop bestaat reeds!	Y	de_DE : missing key
026	nl_NL	spoon	Spoon	Dialog.HopExists.Title	Fout	Y	de_DE : missing key
027	nl_NL	spoon	Spoon	Dialog.ImpactAnalyses.Message	Analyseresultaat:	Y	de_DE : missing key
028	nl_NL	spoon	Spoon	Dialog.ImpactAnalyses.Title	Impactanalyses	Y	de_DE : missing key
029	nl_NL	spoon	Spoon	Dialog.LoadTransformationError.Message	Vanwege een fout kan de transform…	Y	de_DE : missing key
030	nl_NL	spoon	Spoon	Dialog.LoadTransformationError.Title	Fout bij laden transformatie	Y	de_DE : missing key
031	nl_NL	spoon	Spoon	Dialog.NoRepositoryConnection.Message	Er is geen verbinding met de reposit…	Y	de_DE : missing key
032	nl_NL	spoon	Spoon	Dialog.NoRepositoryConnection.Title	Repository niet beschikbaar.	Y	de_DE : missing key
033	nl_NL	spoon	Spoon	Dialog.NoSQLNeedEexecuted.Message	Voorzover het bepaald kan worden, …	Y	de_DE : missing key
034	nl_NL	spoon	Spoon	Dialog.NoSQLNeedEexecuted.Title	SQL	Y	de_DE : missing key
035	nl_NL	spoon	Spoon	Dialog.NotAllowedChangeUser.Message	Sorry, het is u niet toegestaan deze …	Y	de_DE : missing key
036	nl_NL	spoon	Spoon	Dialog.NotAllowedChangeUser.Title	Sorry	Y	de_DE : missing key
037	nl_NL	spoon	Spoon	Dialog.OnlyreadRepository.Message	Sorry, de gebruiker die nu is aangem…	Y	de_DE : missing key
038	nl_NL	spoon	Spoon	Dialog.OnlyreadRepository.Title	Transformation not saved!	Y	de_DE : missing key
039	nl_NL	spoon	Spoon	Dialog.PromptOverwriteFile.Message	Dit bestand bestaat reeds. Wilt u h…	Y	de_DE : missing key
040	nl_NL	spoon	Spoon	Dialog.PromptOverwriteFile.Title	Bestand bestaat reeds!"	Y	de_DE : missing key
041	nl_NL	spoon	Spoon	Dialog.PromptOverwriteTransformation.Message	Er bestaat reeds een transformatie …	Y	de_DE : missing key
042	nl_NL	spoon	Spoon	Dialog.PromptOverwriteTransformation.Title	Overschrijven?	Y	de_DE : missing key
043	nl_NL	spoon	Spoon	Dialog.PromptSave.Message	Het model is gewijzigd. Wilt u de wij…	Y	de_DE : missing key
044	nl_NL	spoon	Spoon	Dialog.PromptSave.Title	Waarschuwing!	Y	de_DE : missing key
045	nl_NL	spoon	Spoon	Dialog.PromptTransformationName.Message	Voorzie de transformatie eerst van e…	Y	de_DE : missing key
046	nl_NL	spoon	Spoon	Dialog.PromptTransformationName.Title	Transformatie heeft geen naam.	Y	de_DE : missing key

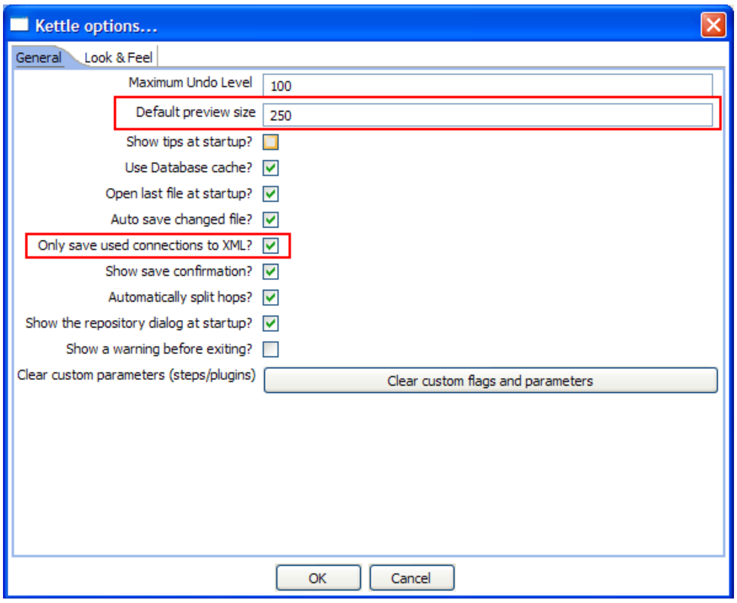
☑ Verify usage ☐ Remove used keys ☑ Check key against other locale

3. Spoon

For a list of the changes that were done in the steps, please see the corresponding chapter below.

3.1. Extra options

These are the new options that were added:

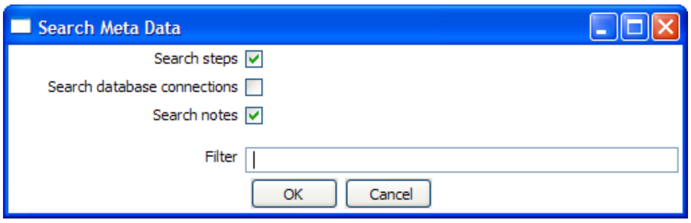


Maximum Undo Level: 100
Default preview size: 250
Show tips at startup? □
Use Database cache? ☑
Open last file at startup? ☑
Auto save changed file? ☑
Only save used connections to XML? ☑
Show save confirmation? ☑
Automatically split hops? ☑
Show the repository dialog at startup? ☑
Show a warning before exiting? □
Clear custom parameters (steps/plugins)

As you can see, it is now possible to set the default number of row to preview. Some people like to see more than others.

The other option will only save the connections that are used in a transformation in XML. This is handy if you have large numbers of connections defined.

3.2. Search meta-data



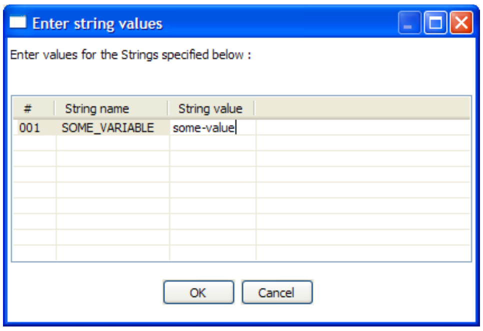
Search steps ☑
Search database connections □
Search notes ☑
Filter

This option will search in the transformation in any possible field, connector or note.

A detailed result is shown.

3.3. Set environment variable

The possibility to set an environment variable was added because it becomes easier that way to test transformations that use dynamically set variables. Normally these variables are set by a different transformation in a job. However, during development and testing, you might want to set these manually.



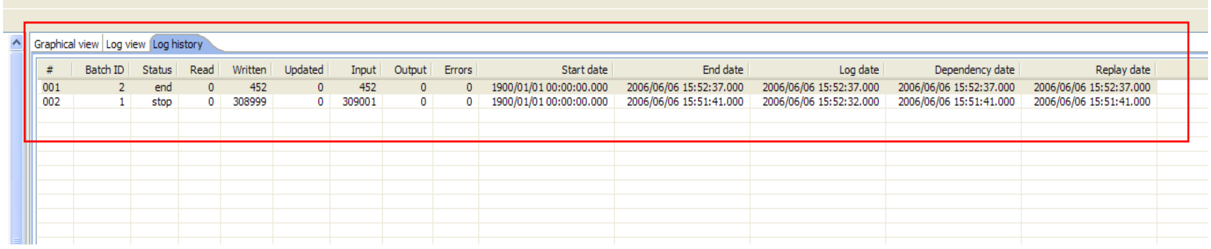
Enter values for the Strings specified below :

#	String name	String value
001	SOME_VARIABLE	some-value

This screen is also presented when you run a transformation that use undefined variables. That way you can define them right before execution time.

3.4. Execution log history

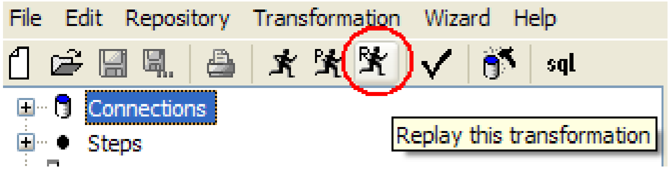
If you store the logging of the transformation in a database table (using the transformation settings, logging tab) you can see the overview of the runs in the log history tab in Spoon:



#	Batch ID	Status	Read	Written	Updated	Input	Output	Errors	Start date	End date	Log date	Dependency date	Replay date
001	2	end	0	452	0	452	0	0	1900/01/01 00:00:00.000	2006/06/06 15:52:37.000	2006/06/06 15:52:37.000	2006/06/06 15:52:37.000	2006/06/06 15:52:37.000
002	1	stop	0	308999	0	309001	0	0	1900/01/01 00:00:00.000	2006/06/06 15:51:41.000	2006/06/06 15:52:32.000	2006/06/06 15:51:41.000	2006/06/06 15:51:41.000

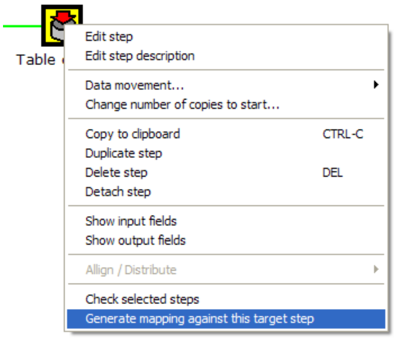
3.5. Replay

It is now possible 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.

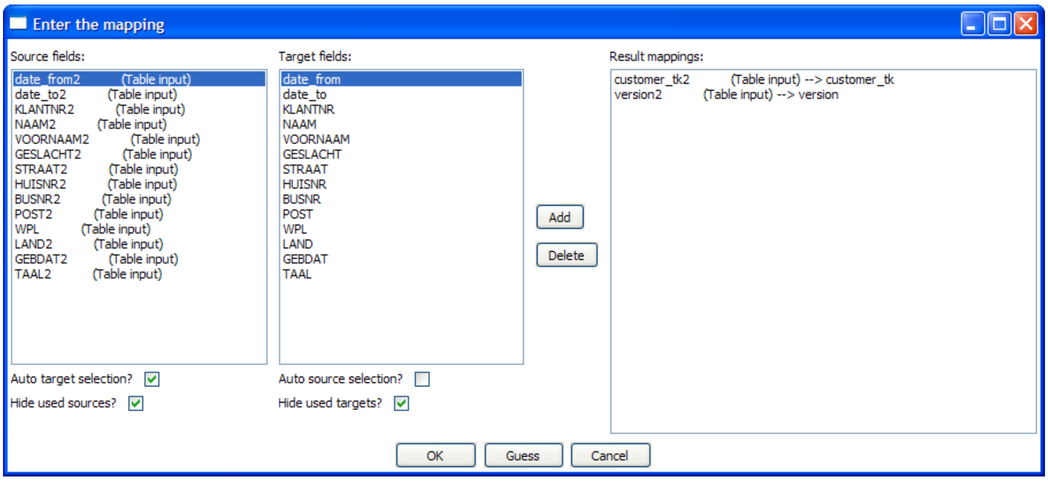


3.6. Generate mapping against target step

In cases where you have a fixed target table, you want to map the fields that go into the table output:



A dialog is then shown that helps you to determine which input fields goes to which table field:



Enter the mapping

Source fields:

```
date_from2     (Table input)
date_to2       (Table input)
KLANTNR2       (Table input)
NAAM2          (Table input)
VOORNAAM2      (Table input)
GESLACHT2      (Table input)
STRAAT2        (Table input)
HUISNR2        (Table input)
BUSNR2         (Table input)
POST2          (Table input)
WPL            (Table input)
LAND2          (Table input)
GEBDAT2        (Table input)
TAAL2          (Table input)
```

Target fields:

```
date_from
date_to
KLANTNR
NAAM
VOORNAAM
GESLACHT
STRAAT
HUISNR
BUSNR
POST
WPL
LAND
GEBDAT
TAAL
```

Result mappings:

```
customer_tk2     (Table input) --> customer_tk
version2         (Table input) --> version
```

Auto target selection? ☑

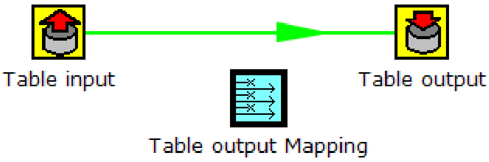
Hide used sources? ☑

Auto source selection? □

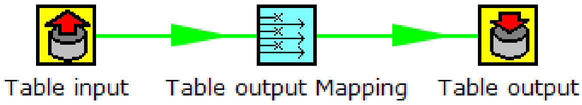
Hide used targets? ☑

The selection of the target fields are done semi-automatic based on (parts of) the field name.

As a result of this dialog, a new Select Values step is generated:



We can simply place this step between the two and the mapping from input to output is done:



3.7. 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. 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 safemode option.*

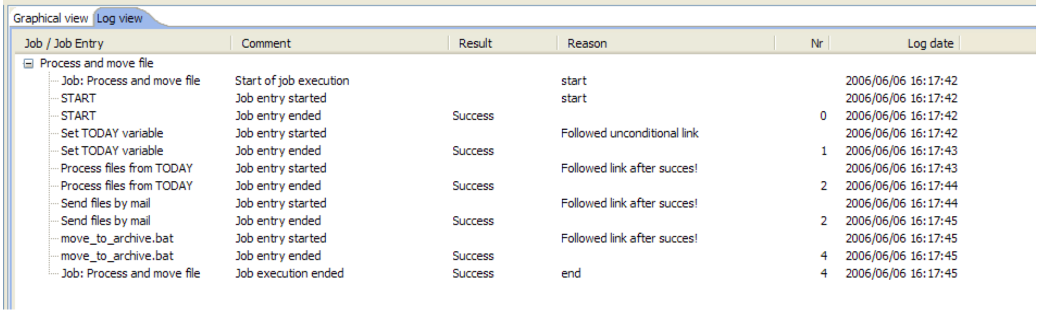
4. Chef

Although it may appear at first glance that not too much has changed, a lot of things were fixed and improved upon in version 2.3.0.

For a list of the changes that were done in the job entries, please see the corresponding chapter below.

4.1. New logging window

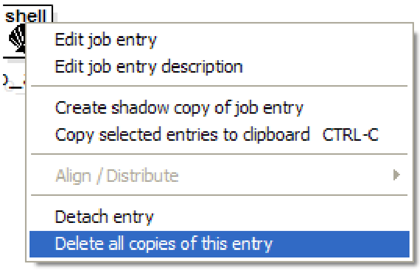
As you can see below, the execution of the job entries is now shown in a hierarchical way using a tree. This allows you to track the processing more easily.



Job / Job Entry	Comment	Result	Reason	Nr	Log date
Process and move file					
Job: Process and move file	Start of job execution		start		2006/06/06 16:17:42
START	Job entry started		start		2006/06/06 16:17:42
START	Job entry ended	Success		0	2006/06/06 16:17:42
Set TODAY variable	Job entry started		Followed unconditional link		2006/06/06 16:17:42
Set TODAY variable	Job entry ended	Success		1	2006/06/06 16:17:43
Process files from TODAY	Job entry started		Followed link after succes!		2006/06/06 16:17:43
Process files from TODAY	Job entry ended	Success		2	2006/06/06 16:17:44
Send files by mail	Job entry started		Followed link after succes!		2006/06/06 16:17:44
Send files by mail	Job entry ended	Success		2	2006/06/06 16:17:45
move_to_archive.bat	Job entry started		Followed link after succes!		2006/06/06 16:17:45
move_to_archive.bat	Job entry ended	Success		4	2006/06/06 16:17:45
Job: Process and move file	Job execution ended	Success	end	4	2006/06/06 16:17:45

4.2. Delete job entry copies

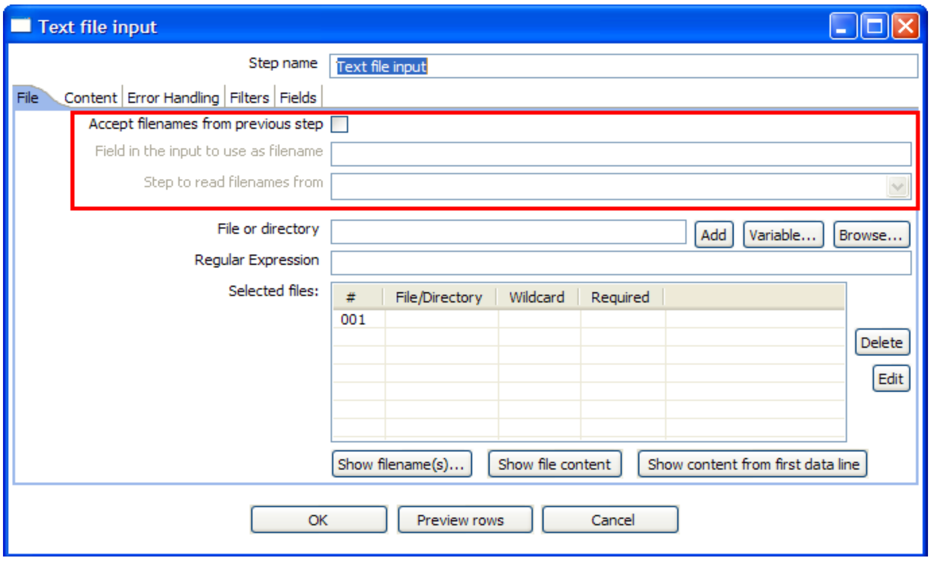
To allow you to delete created job entries faster, you now also have a new "Delete all copies off this entry" feature.


shell
Edit job entry
Edit job entry description
Create shadow copy of job entry
Copy selected entries to clipboard CTRL-C
Align / Distribute
Detach entry
Delete all copies of this entry

5. Steps

5.1. Text file input

- Accept filenames from previous steps, together with the new step "Get file names" you can compose your own filenames, or get filenames from text-files, databases, etc.



Step name: Text file input

Accept filenames from previous step ☐

Field in the input to use as filename

Step to read filenames from

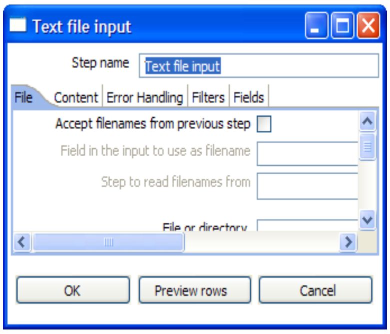
File or directory

Regular Expression

Selected files:

#	File/Directory	Wildcard	Required
001			

- Scroll-bars appear in this step to prevent options from being hidden



Step name: Text file input

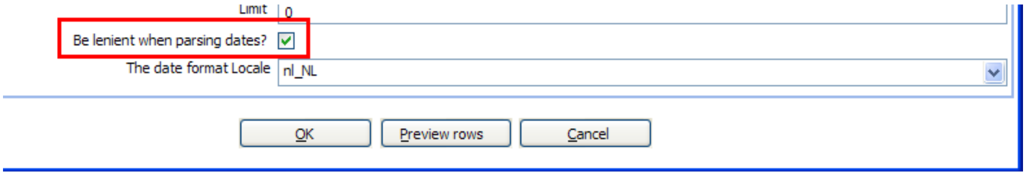
Accept filenames from previous step ☐

Field in the input to use as filename

Step to read filenames from

File or directory

- Date parsing can be set to "lenient" or strict. In lenient mode, February 31st is converted to the corresponding day in March. In not in lenient mode, an error is thrown.

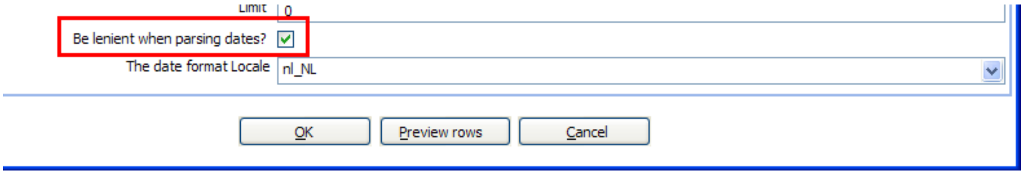


Limit 0

Be lenient when parsing dates? ☑

The date format Locale nl_NL

- The locale can be set for date formats, this is needed when parsing dates written in full in different locale:

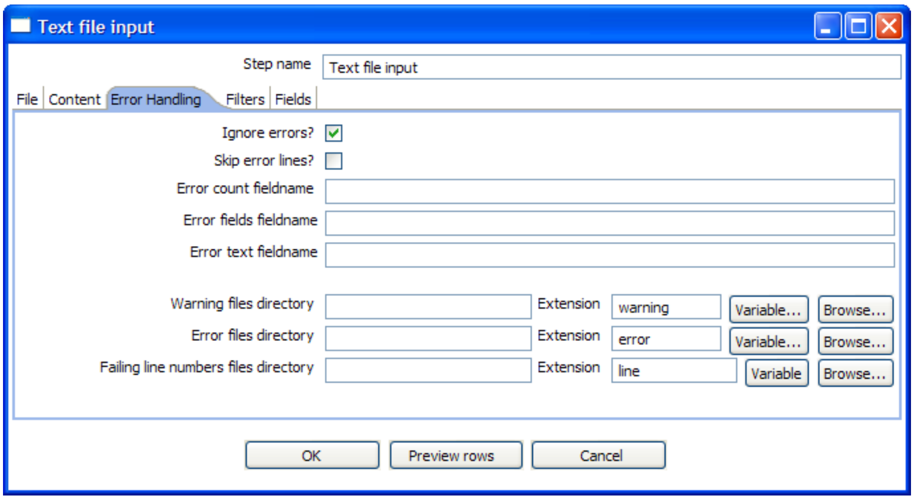


Limit 0

Be lenient when parsing dates? ☑

The date format Locale nl_NL

- Then we added a whole list of extra features to make error handling better for automated text-file handling:



Step name Text file input

Ignore errors? ☑

Skip error lines? □

Error count fieldname

Error fields fieldname

Error text fieldname

Warning files directory Extension warning

Error files directory Extension error

Failing line numbers files directory Extension line

These features allow you to create ignore errors, send the failing field names to next steps or even create files with line numbers in it on which the error(s) occurred. When the transformation is replayed, the other lines can be ignored.

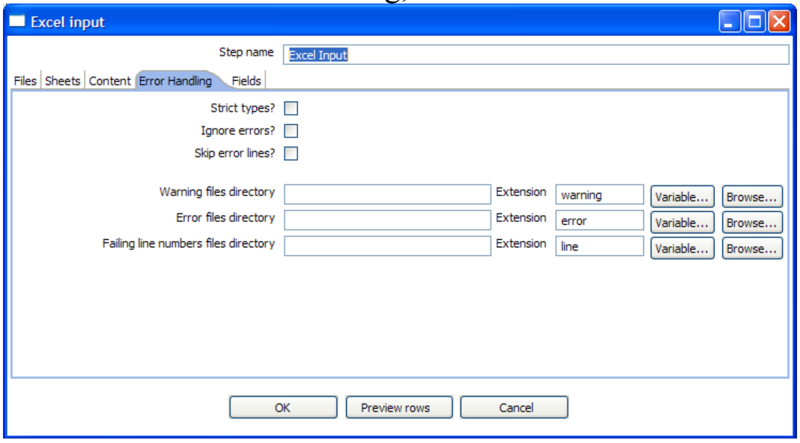
5.2. Get System Info

Even though at first glance nothing has changed here, it is now more convenient to add values from this step to a stream by simply giving this step input. The selected values will be added to the output.

5.3. Excel Input

As reading excel files is something that's used a lot, this step got some attention as well.

- First there is the error handling, modeled after the Text File Input step:

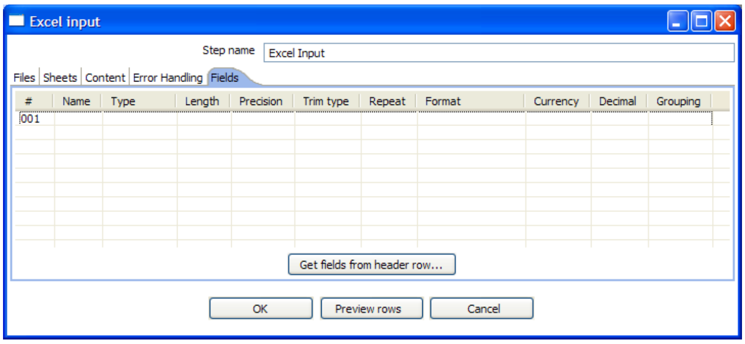


Step name: Excel Input

Strict types? ☐
Ignore errors? ☐
Skip error lines? ☐

Warning files directory: Extension: warning
Error files directory: Extension: error
Failing line numbers files directory: Extension: line

- Because any data type can be found in a cell in a sheet in a worksheet, you need some kind of conversion logic. This is covered in the new Fields tab:



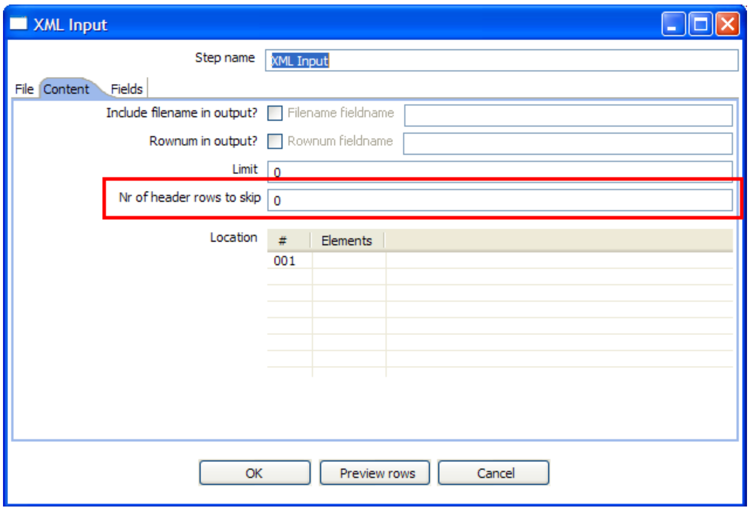
Step name: Excel Input

#	Name	Type	Length	Precision	Trim type	Repeat	Format	Currency	Decimal	Grouping
001										

- Last but not least, Sven Boden added support for Formula fields. As it seems, the result of the formula is stored in the Excel file and you grab it without too much problems.

5.4. XML Input

- While reading excel files, stored in XML, James Dixon wanted to skip the header row(s). So this option was added just for that:



Step name: XML Input

Include filename in output? □ Filename fieldname

Rownum in output? □ Rownum fieldname

Limit: 0

Nr of header rows to skip: 0

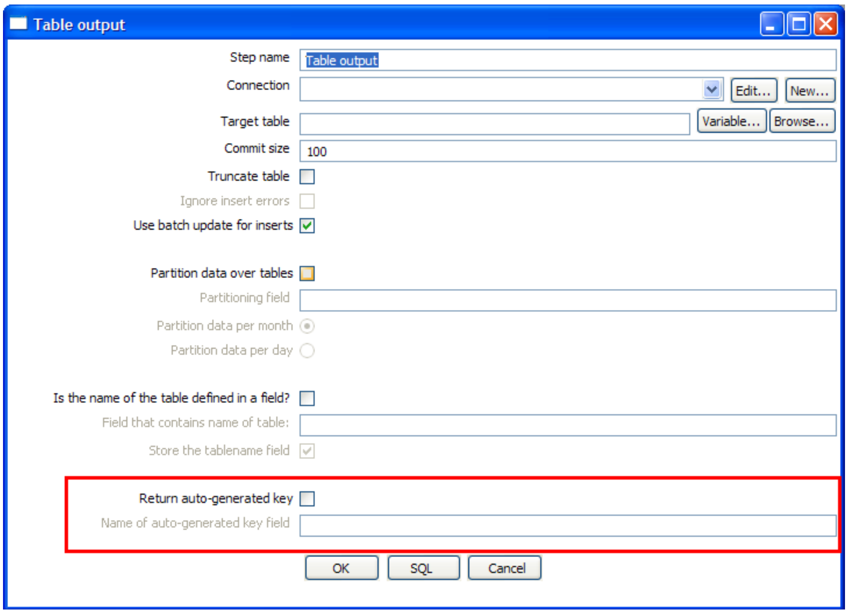
Location

#	Elements
001	

- Support was added for grabbing information from XML documents without repeating elements. In that case, you need to specify a single Element in "Location".
- 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.

5.5. Table output

- When a row of data is written to a table, and a new key is generated in an auto-increment field, you might want to grab that generated key. You can do so now with this new feature:



Step name: Table output
Connection:
Target table:
Commit size: 100
Truncate table □
Ignore insert errors □
Use batch update for inserts ☑

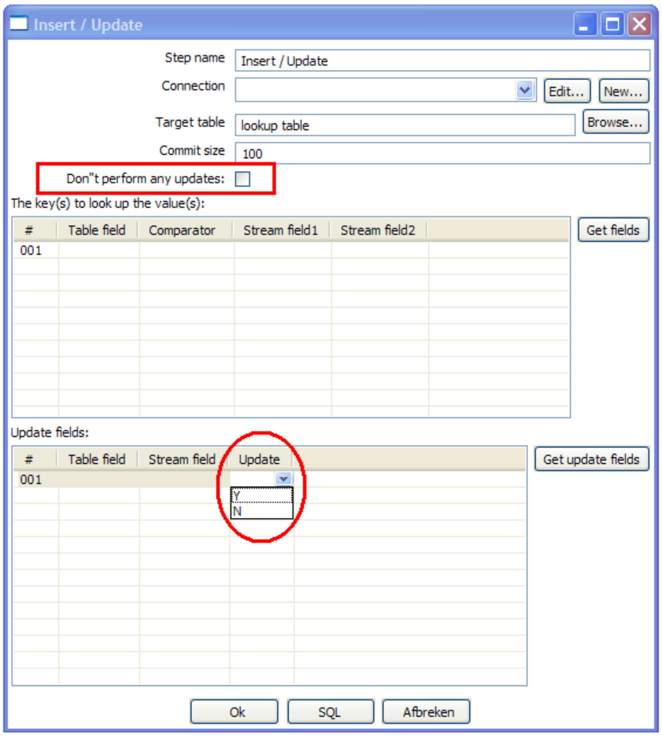
Partition data over tables □
Partitioning field
Partition data per month ●
Partition data per day ○

Is the name of the table defined in a field? □
Field that contains name of table:
Store the tablename field ☑

Return auto-generated key □
Name of auto-generated key field

5.6. Insert / Update

- Sometimes you don't want to perform an update in a database table if the row is already present. That's the reason this option was added:



Step name: Insert / Update
Connection:
Target table: lookup table
Commit size: 100
Don"t perform any updates: □

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
001				

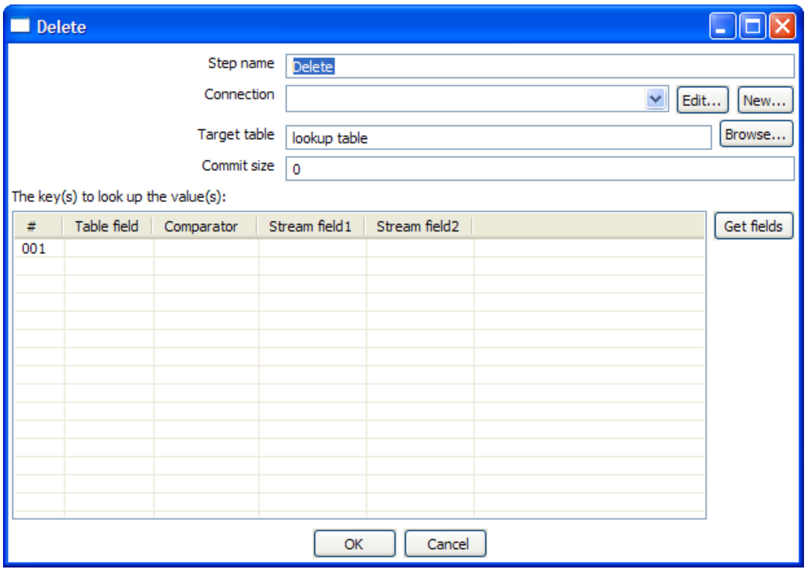
Update fields:

#	Table field	Stream field	Update
001			Y N

- The updates in the table can now also be specified on a field level.
- A number of smaller performance updates were done to this step as well. Also, a serious dead-lock issue was removed on certain databases because this step used more than 1 connection to do inserts, updates and lookups. If the same key passed more than once, this caused dead-locks in 2.2.2

5.7. Delete (new step)

- The delete step allows you to delete data from a database table:



Delete

Step name: Delete

Connection:

Target table: lookup table

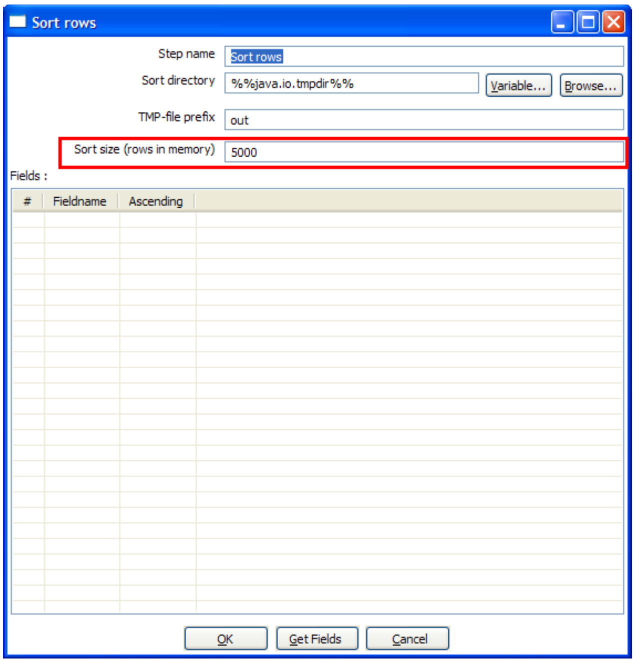
Commit size: 0

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
001				

5.8. Sort rows

- To speed up the sorting, you can specify the number of rows in the buffer, the more rows you can keep in the buffer, the faster sorting gets (it reduces I/O):



Step name: Sort rows

Sort directory: %%java.io.tmpdir%%

TMP-file prefix: out

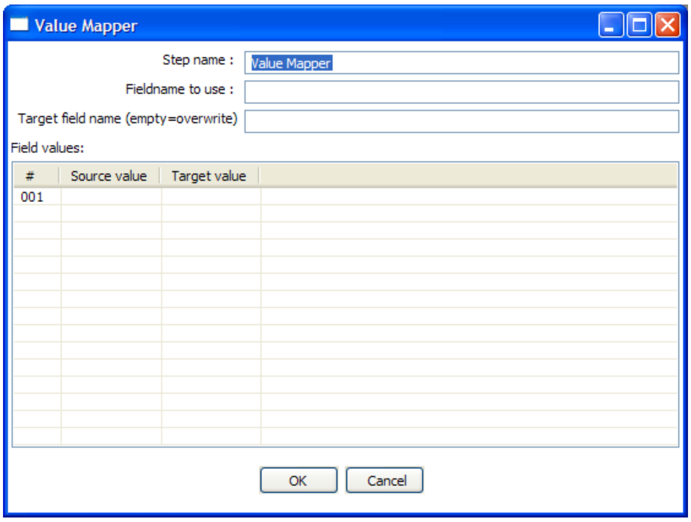
Sort size (rows in memory): 5000

Fields :

#	Fieldname	Ascending

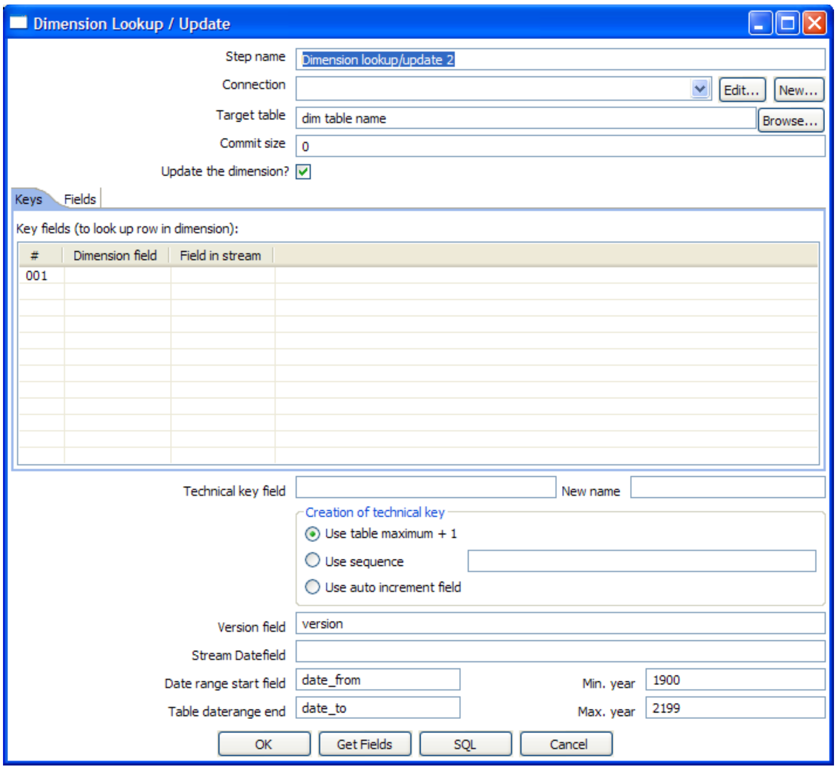
5.9. Value Mapper (New step)

- If you simply want to change a certain value to a different string value:



5.10. Dimension Lookup

- Mostly GUI changes here, trying to make it easier to use:



Dimension Lookup / Update

Step name: Dimension lookup/update 2

Connection:

Target table: dim table name

Commit size: 0

Update the dimension? ☑

Keys | Fields

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
001		

Technical key field: New name:

Creation of technical key

● Use table maximum + 1

○ Use sequence

○ Use auto increment field

Version field: version

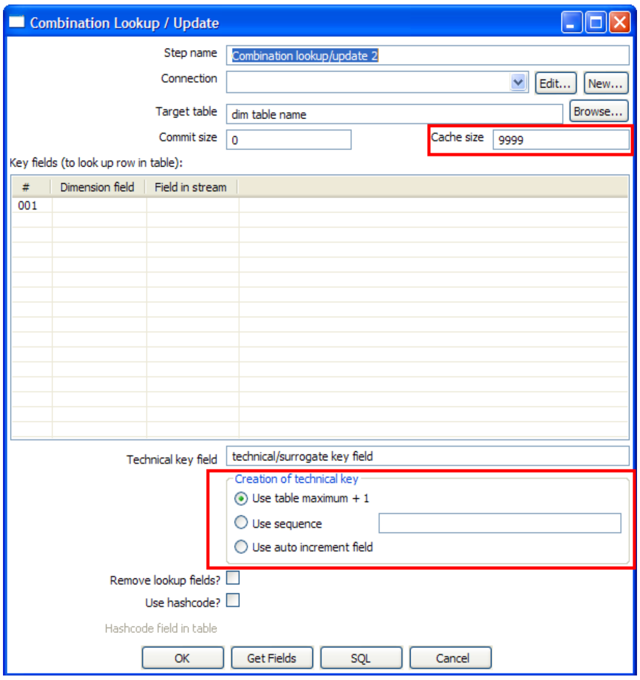
Stream Datefield:

Date range start field: date_from Min. year: 1900

Table daterange end: date_to Max. year: 2199

5.11. Combination Lookup

- Mostly GUI changes here, trying to make it easier to use.
- Making support for auto-increment fields more configurable.
- Support for caching rows of data.



Combination Lookup / Update

Step name: Combination lookup/update 2

Connection:

Target table: dim table name

Commit size: 0

Cache size: 9999

Key fields (to look up row in table):

#	Dimension field	Field in stream
001		

Technical key field: technical/surrogate key field

Creation of technical key

● Use table maximum + 1

○ Use sequence

○ Use auto increment field

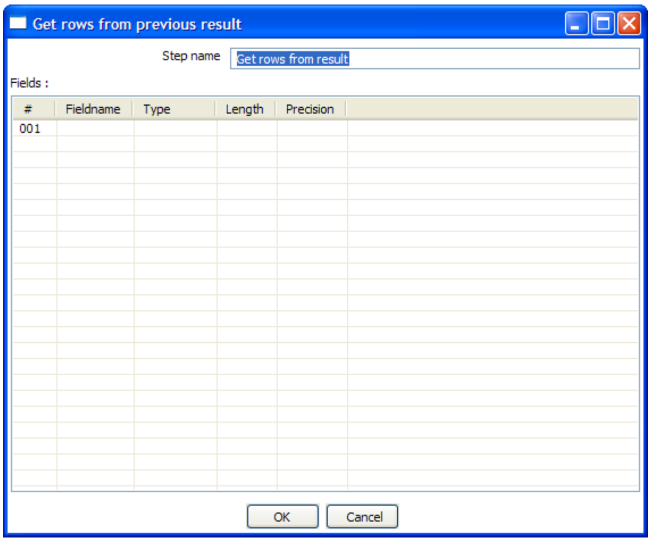
Remove lookup fields? □

Use hashcode? □

Hashcode field in table

5.12. Get rows from result

- To make it easier to specify the meta-data of rows rows of data from previous transformations / jobs, you can simply specify the fields you are expecting. A Nicholas Goodman idea:

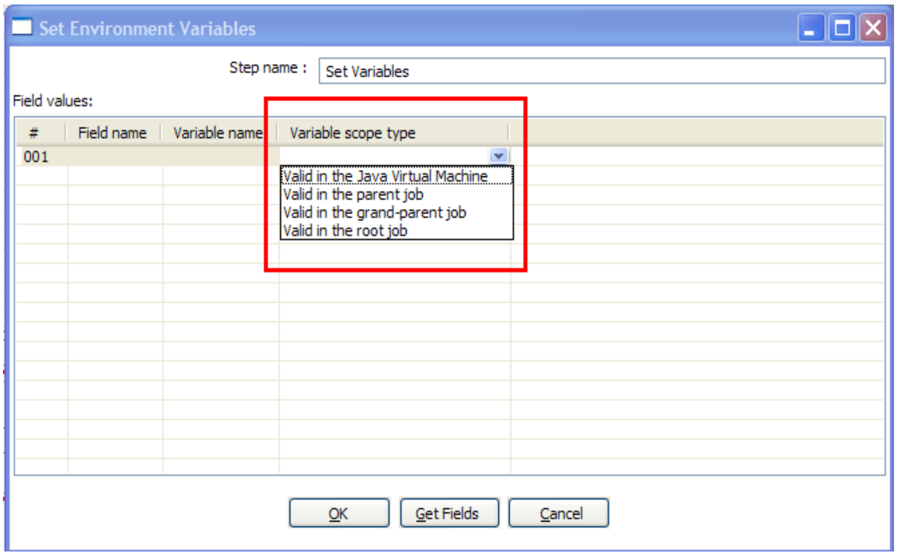


Step name: Get rows from result

#	Fieldname	Type	Length	Precision
001				

5.13. Set Variable (New step)

- Because of the dynamic nature of certain requirements (deployments etc), you might want to set variables also dynamically in a job. You now can do this with the Set Variables step:



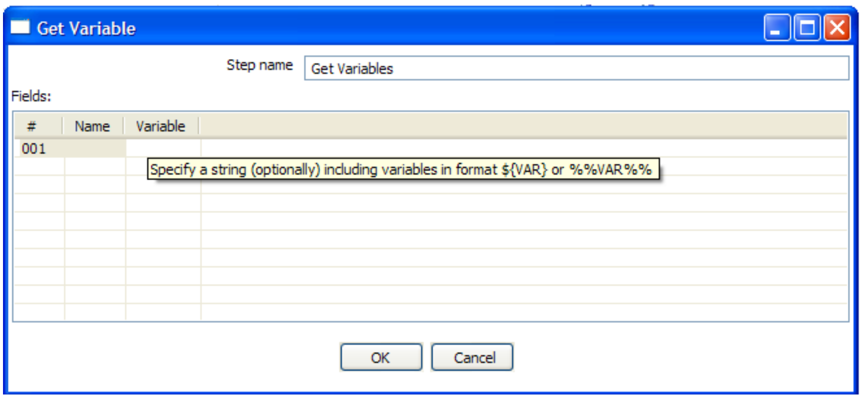
Step name : Set Variables

001

Valid in the Java Virtual Machine
Valid in the parent job
Valid in the grand-parent job
Valid in the root job

5.14. Get Variable (New step)

- If you want to add information from a variable to a stream of data:



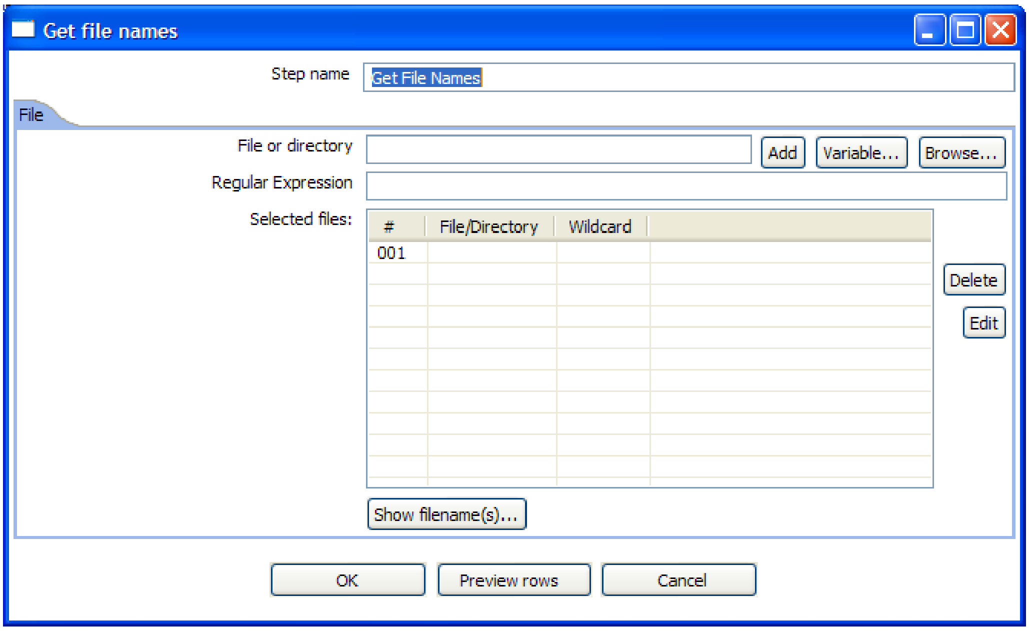
Step name Get Variables

001

Specify a string (optionally) including variables in format ${VAR} or %%VAR%%

5.15. Get file names (New step)

- File names are information too, so in case you want to grab a filename, just use this step:



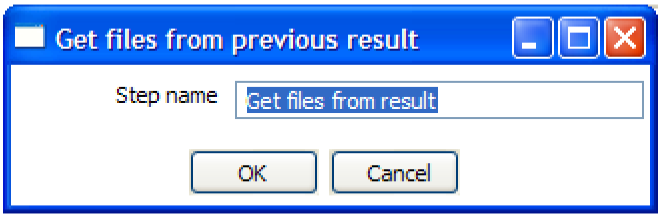
Get file names

Step name: Get File Names

- This code was "stolen" from Text File Input and works just the same. With the except that the generated filenames are simply sent to the next steps in the field named "filename".
- You can use this to send the file names directly to Text File Input *or* (a more dynamic approach) send the filenames to the next "Job" job entry (in a Job in Chef) that sets a variable "FILENAME" and then you can process the file, move the file to an archive directory, mail the file, etc. This kind of functionality is very useful to replace shell scripting. Those have typically a high maintenance cost and are often platform specific.

5.16. Get files from result (New step)

- 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:



Get files from previous result

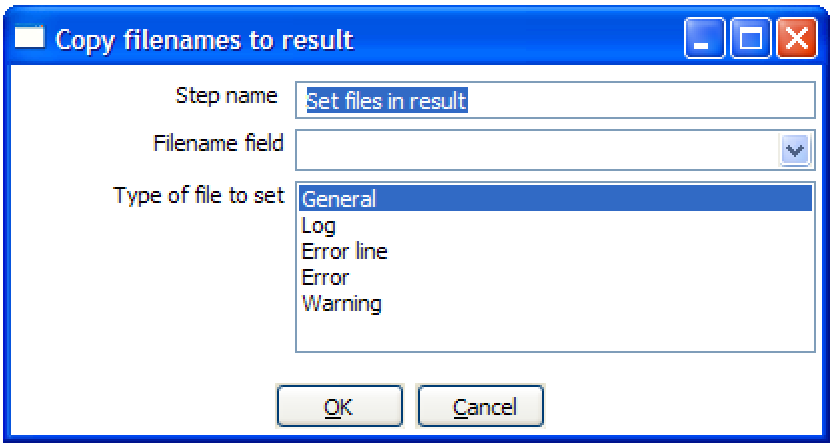
Step name: Get files from result

- 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

5.17. Set files in result (New step)

- So, in certain cases, to steer the list of files in the result ourself, we can use this step:



Copy filenames to result

Step name: Set files in result

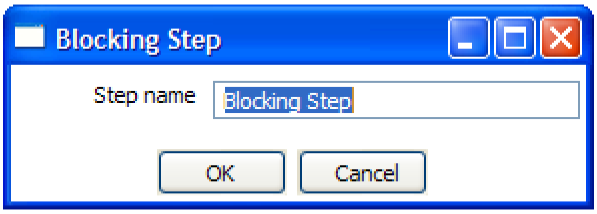
Filename field:

Type of file to set: General (selected), Log, Error line, Error, Warning

- 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.

5.18. Blocking step (New step)

- This is again a very simple step. It blocks all output until the very last row was received from the previous step. This last row is then sent off to the next step. This step then nows that all previous steps have finished. You can use this for triggering custom plugin, stored procedures, java scripts, etc. :

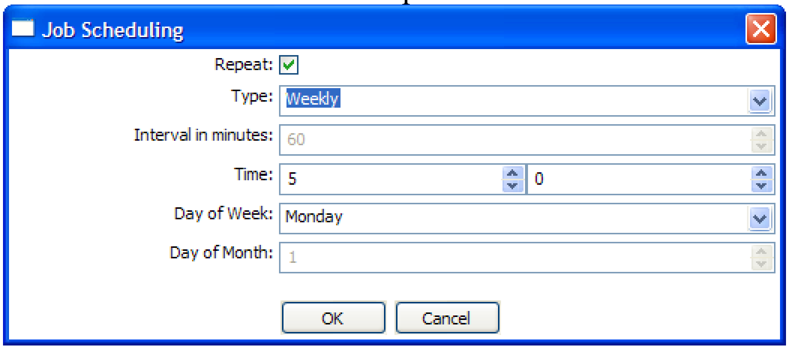


Step name: Blocking Step

6. Job Entries

6.1. Start

- You can set the schedule of the job by simply specifying until when the "Start" job entry should block or if it needs to repeat:



Job Scheduling

Repeat: ☑

Type: Weekly

Interval in minutes: 60

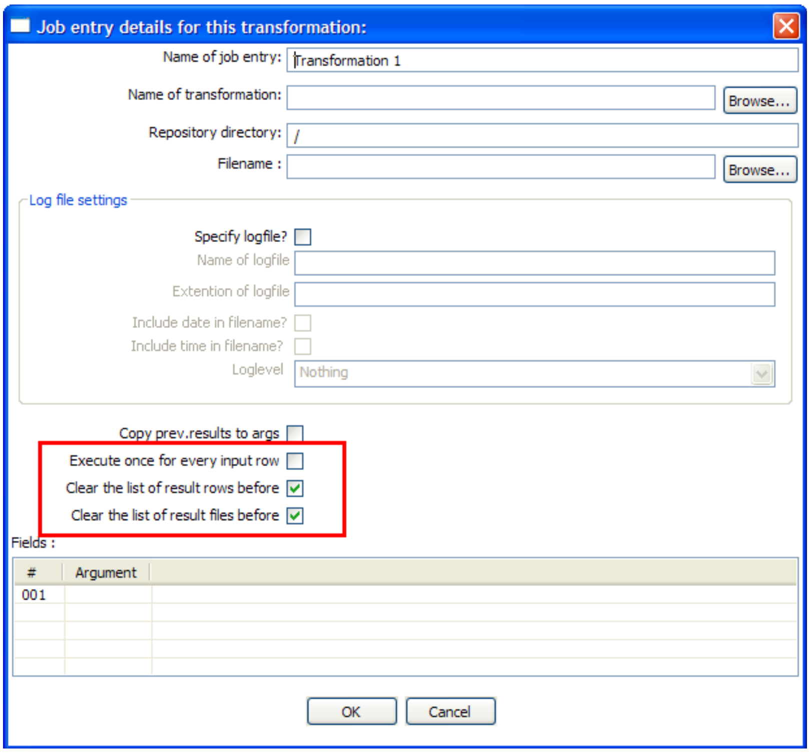
Time: 5 0

Day of Week: Monday

Day of Month: 1

6.2. Transformation

- Support for "looping" has been added by allowing a transformation to be executed once for every input row:



Job entry details for this transformation:

Name of job entry: Transformation 1

Name of transformation:

Repository directory: /

Filename :

Log file settings

Specify logfile? ☐

Name of logfile

Extention of logfile

Include date in filename? ☐

Include time in filename? ☐

Loglevel Nothing

Copy prev.results to args ☐

Execute once for every input row ☐

Clear the list of result rows before ☑

Clear the list of result files before ☑

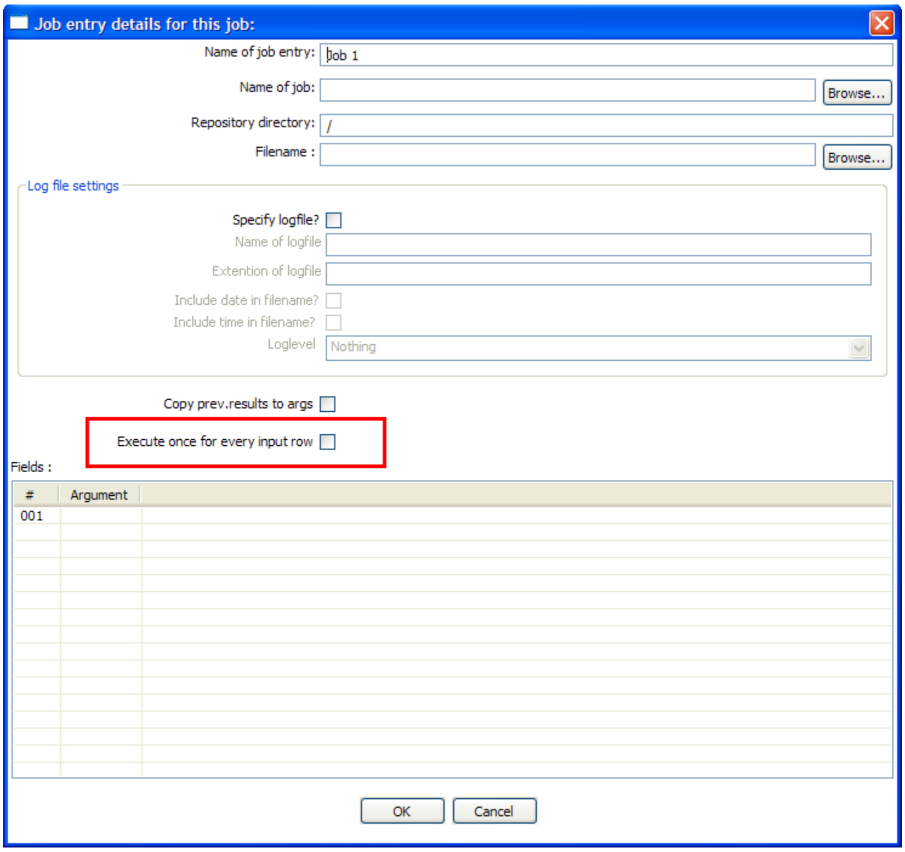
Fields :

#	Argument
001	

- The (file-) name of the transformation can be specified using variables as well. This allows for easier deployment logic.

6.3. Job

- Support for “looping” has been added by allowing a job to be executed once for every input row:



Job entry details for this job:

Name of job entry: Job 1

Name of job:

Repository directory: /

Filename :

Log file settings

Specify logfile? □

Name of logfile

Extention of logfile

Include date in filename? □

Include time in filename? □

Loglevel: Nothing

Copy prev.results to args □

Execute once for every input row □

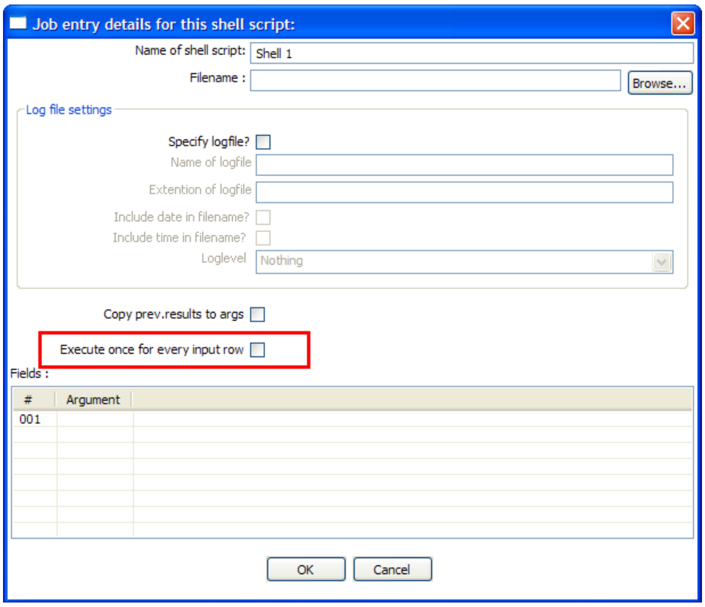
Fields :

#	Argument
001	

- The (file-) name of the job can be specified using variables as well. This allows for easier deployment logic.

6.4. Shell

- Support for “looping” has been added by allowing a shell script to be executed once for every input row:



Job entry details for this shell script:

Name of shell script: Shell 1

Filename :

Log file settings

Specify logfile? □

Name of logfile

Extention of logfile

Include date in filename? □

Include time in filename? □

Loglevel Nothing

Copy prev.results to args □

Execute once for every input row □

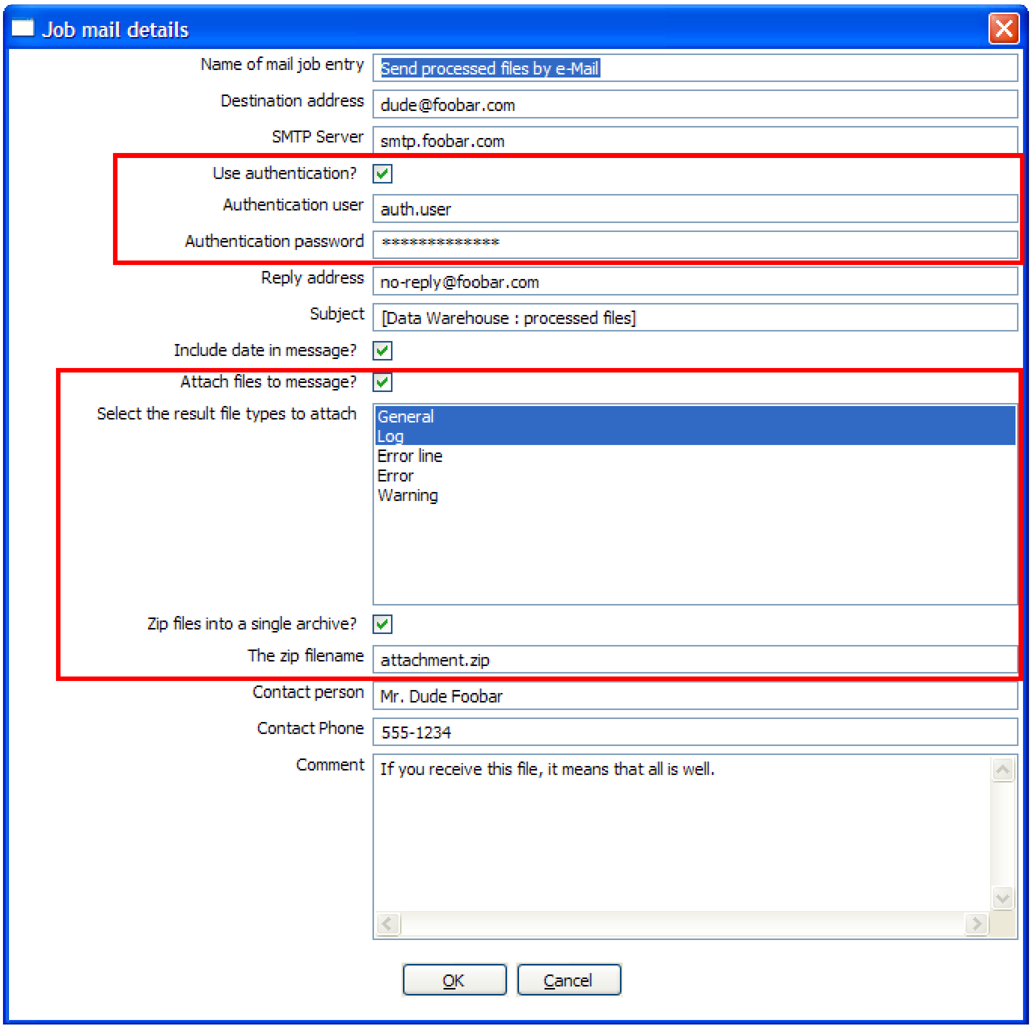
Fields :

#	Argument
001	

- Also, output generated by the shell script (console output and error output) are captured and sent to the logging. This no longer blocks the execution of the job entry.
- On windows, scripts are now preceded by “CMD.EXE /C” (NT/XP/2000) or “COMMAND.COM /C” (95,98).
- The (file-) name of the shell script can be specified using variables as well. This allows for easier deployment logic.

6.5. Mail

- Support for authentication has been added, as well as (zipped) attachments.



Name of mail job entry: Send processed files by e-Mail

Destination address: dude@foobar.com

SMTP Server: smtp.foobar.com

Use authentication? ☑

Authentication user: auth.user

Authentication password: **************

Reply address: no-reply@foobar.com

Subject: [Data Warehouse : processed files]

Include date in message? ☑

Attach files to message? ☑

Select the result file types to attach: General, Log, Error line, Error, Warning

Zip files into a single archive? ☑

The zip filename: attachment.zip

Contact person: Mr. Dude Foobar

Contact Phone: 555-1234

Comment: If you receive this file, it means that all is well.

- The full trace of the job execution prior to arriving at the mail job entry is also included in the mail as well as the content of the result. (number of lines read, written, etc.)
- All fields can be specified using environment variables as well. For example: ${DESTINATION_ADDRESS}
- Specify multiple destination addresses separated with spaces.

7. Source code improvements

7.1. A few extra lines of code

Version 2.2.2 contains 177,450 lines of code.

Version 2.3.0 contains 213,489 lines of code, an increase of 36,039 lines.

7.2. Trackers

In the period between the release of version 2.2.2 until June 6th, 112 bugs were fixed and 21 feature requests from users were implemented.

7.3. Unit testing

To guard quality and backward compatibility in this release and for the future ones, we started with the implementation of several unit tests. Sven Boden, fellow Belgian is implementing tests for all core classes and has so far covered 5% of the complete Kettle code base.

7.4. Commiters

ID / Commits	e-Mail	Name	Country	Work
audinchan 10	Audinchan (at) gmail.com	Chan Audin	CH	Repository importer fixes Various Text File Input fixes
berarma 129	Bernardo (at) tsolucio.com	Arlandis Bernardo	ES	A lot of i18n fixes. Various postgres database fixes Better architecture detection in .sh scripts
DennisVR 11	Dennis (at) iqit.be	Dennis Van Roeyen	B	Basic scheduler for Jobs Insert/update step: update field based
hoezx6r 40	Ahoesley (at) skyroadasp.com	Andrew Hoesley	USA	Reworked Job Entry Plugin system Fixed plugin loading actually Text File Input: added default for null values Text File Input: file handling for jobs
jbleuel 132	Jens.bleuel (at) proratio.de	Jens Bleuel	D	Database wizard fixes Database reserved words: DB2, AS/400, MS-Access, MS-SQL Server, Postgres Various bug fixes German translations and i18n fixes
jvanhent 455	johnny.vanhentenryk (at) ixor.be	Johnny Vanhentenryk	B	Error handling for Text File Input / Excel Input as well as replay functionality, various fixes, data lenient (TFI), log window history refresh...
qinhui99 472	Qinhui99 (at) hotmail.com	Tom Qin	CH	String externalization Translation into Simplified Chinese Various bug fixes
sboden 120	Svenboden (at) hotmail.com	Sven Boden	B	Wrote lots of Unit tests Code review, verifications & hardening Dimension & Combination lookup improvements and GUI changes Various bug fixes
vitoecn	Through: Qinhui99 (at)	Iv Vitoe	CH	Various i18n improvements

ID / Commits	e-Mail	Name	Country	Work
	hotmail.com			Translation into Simplified Chinese
wconroy 31	Wconroy (at) gmail.com	Bo Conroy	USA	New Blocking step FTP: allow variables for regex and target dir. fix timeout. Added ValueSerializable data type Various bug fixes Java advice and proposals
wdeclerc 47	Wim.DeClercq (at) ixor.be	Wim De Clercq	BE	Parameterizing database connections Implementation of kettle.properties file General java advice
MattCasters 2,434	Mcasters (at) pentaho.org	Matt Casters	BE	Various stuff
TOTAL 3,919	***12 developers***		***6 countries***	

7.5. Contributors

TODO