
# Pentaho Data Integration version 2.5.0

## Getting the job done

*Key differences with Kettle 2.4.0*

*List of changes on 20-04-'07*

*Compiled by Matt Casters, mcasters (at) pentaho.org*

*Send additional changes you found to this address.*

# Index

# 1. Changes summary

## 1.1. Preface

It was only in early February that we released the previous version of Pentaho Data Integration, version 2.4.0. Originally our plan was to just release an updated point release (2.4.1). However, we received so many contributions and added really cool features that we had to go for a major release anyway.

Although we did gain a lot of new job entries to help you do a better "job" in the work flow department, very exciting changes where also done in the transformation engine. As you can see below, advanced error handling and Apache Virtual File System support are great additions to our software.

This document was written as a special "thank you" note to all people involved in the community and to keep everyone informed about the incredible progress we are making.

## 1.2. Overview

These are the most notable changes that have been made:

- Advanced error handling added
  - Allowing rows of data that cause an error to be re-routed
  - Allows for cleaner transformations
  - Doesn't slow the transformation down at all
  - Offers interesting new possibilities for quality control and update strategies

- Included Apache VFS support
  - Allows you to read/write files directly from URLs
  - More info is here: http://jakarta.apache.org/commons/vfs/

- Re-designed the left trees
  - Turned them into an easier to use toolbar
  - Reduces lookup time for steps and objects
  - Added a new Favorite steps section

- New Steps
  - Abort step : halts a transformation in case one or more records enter the step. This can be used in association with the new error handling capabilities
  - Formula (experimental)
  - Web services lookup (experimental)

- New Job entries
  - Create file
  - Delete file
  - Wait for file
  - Put a file with Secure FTP (STFP)
  - File compare
  - Bulk load into MySQL
  - Display Msgbox info
  - Wait for
  - Zip file
  - BulkLoad from MySQL into file
  - Abort job
  - Get mails from POP
  - Ping a host

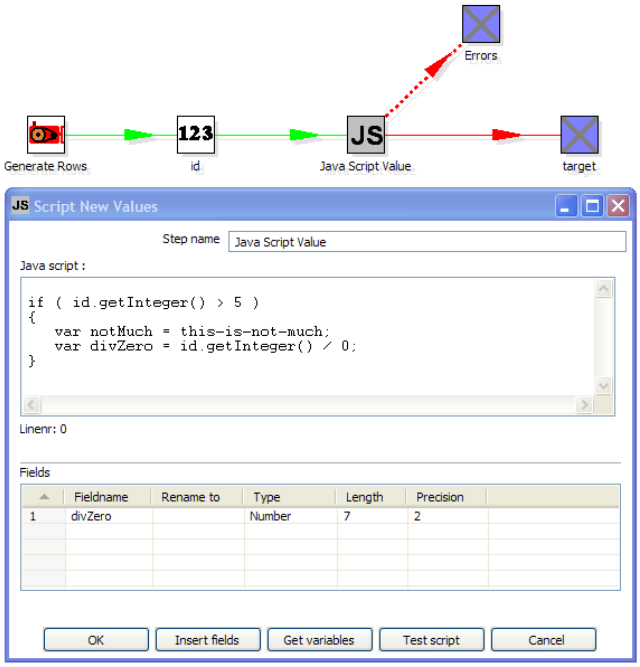
- Miscellaneous
  - Hundreds of bugs fixed and dozens of change requests implemented
  - Made the Modified Javascript more compatible with the old Javascript engine.
  - Added caching to the Dimension Lookup/Update step
  - Lots of internationalization efforts took place, for the welcome page, screens and manuals.
  - Easier to create new files
  - Reduced clutter, improved UI usability of Spoon
  - Give various warnings when mixing row layouts at design time
  - >500.000 download attempts of 2.4.0 (including a few DOS attacks)
  - ...

# 2. General changes

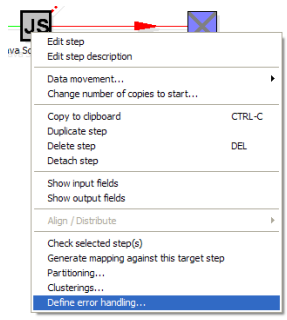
## 2.1. Advanced error handling

This feature originated as a great idea from Sven Boden, one of the core developers of Pentaho Data Integration. The idea was simple: in stead of halting a transformation when an error occurs in a step, you should be able to pass those rows that cause an error to a different step.

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

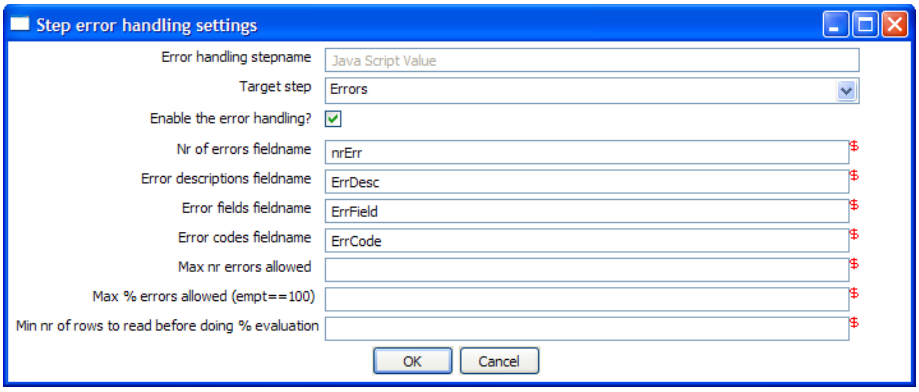



To configure the error handling, you can right click on the step involved and select the “Error handing...” menu item:

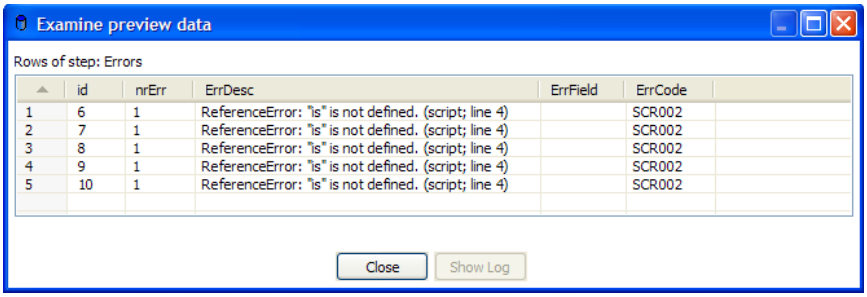



NOTE: this menu item only appears when clicking on steps that support the new error handling code.

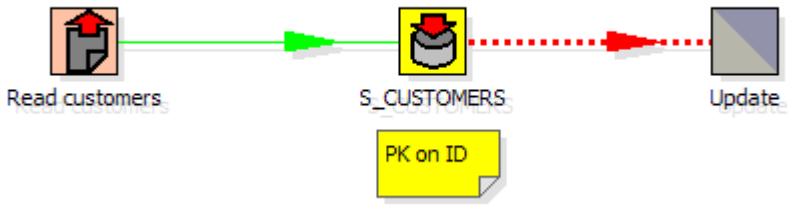
The error handling dialog looks like this:




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




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




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

## 2.2. Apache VFS support

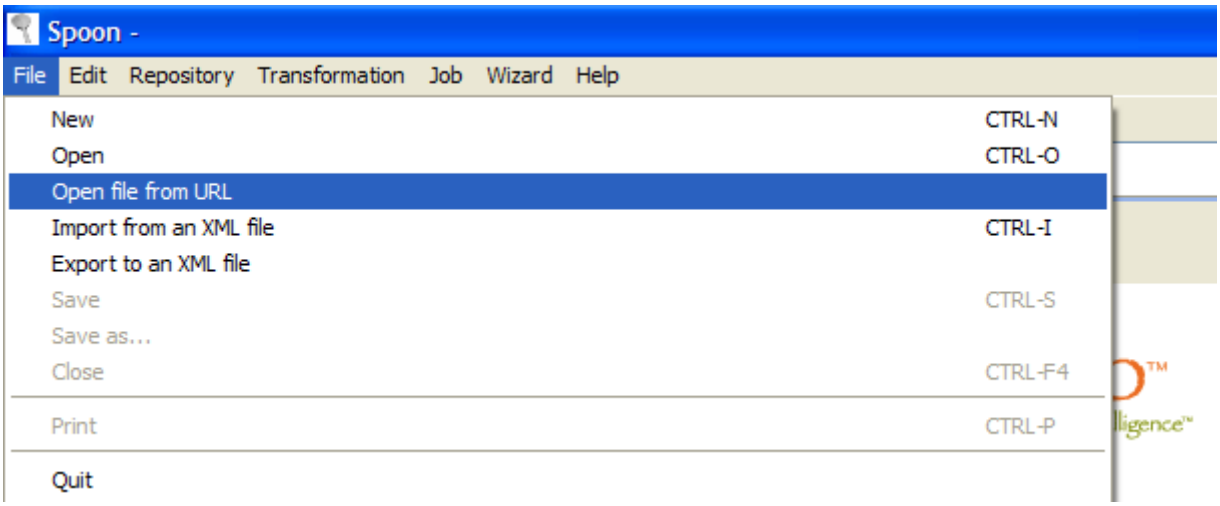
One of the new cool things that we recently implemented is the ability to reference source files, transformations and jobs from any location you like.

The underlying libraries we use to do that is the Apache Commons Virtual File System.

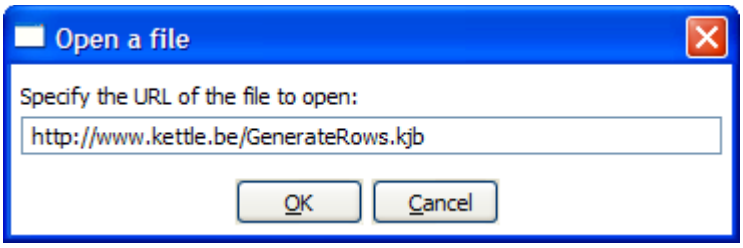
Here is a simple example that you can try with the latest dev version:

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

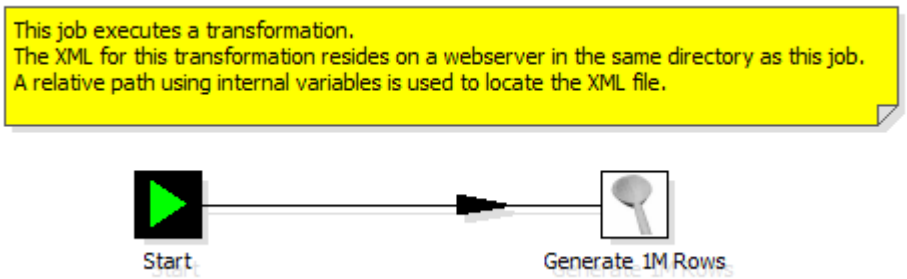
Let's have a look at this job in Spoon. To open it directly from the URL above follow this procedure:




Type in the URL:



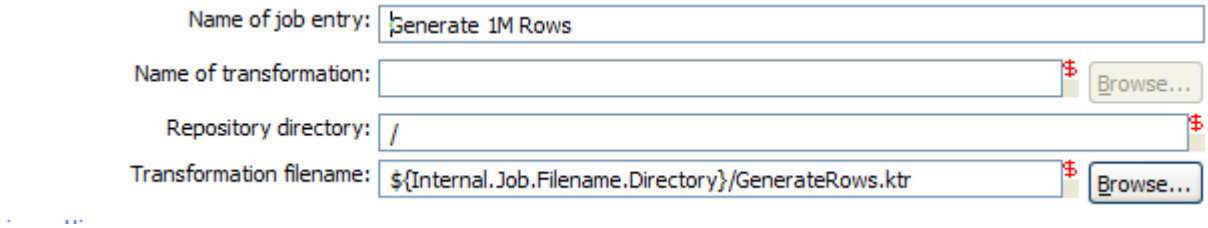

Selecting OK will load the job in Spoon:



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

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

This allows us to reference the transformation as follows:

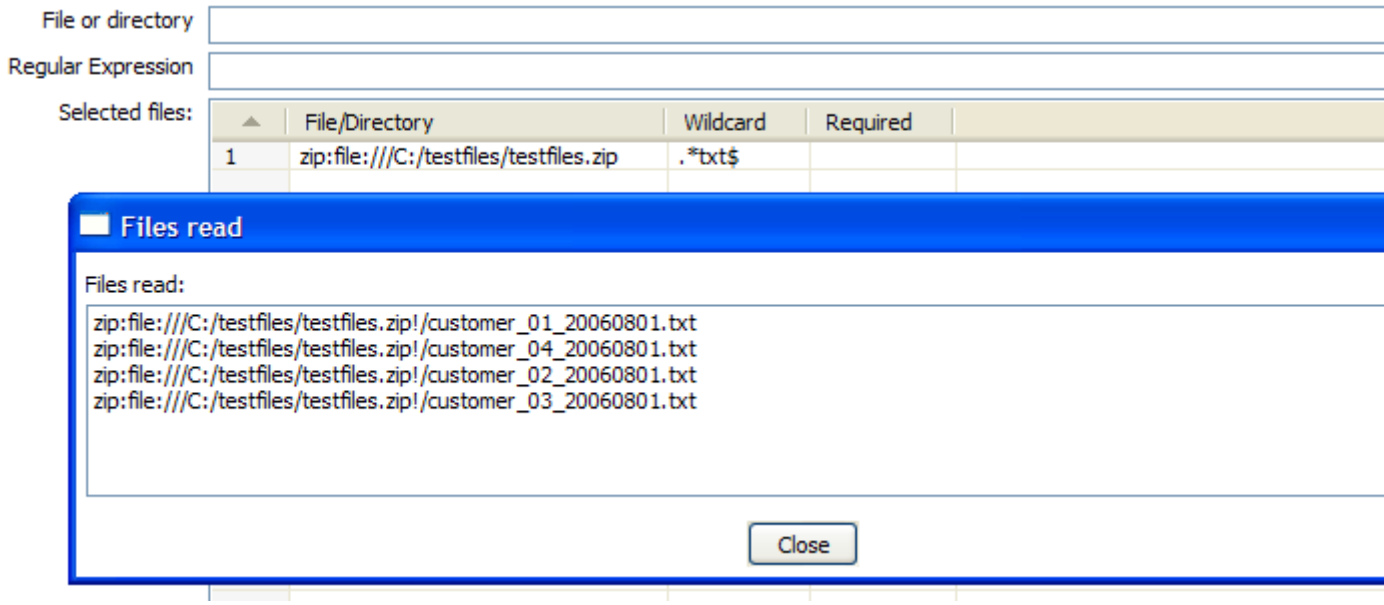



Please note that if you try this yourself you'll note that you can't save the job back to the webserver. That is not because we don't support that, but because you don't have the permission to so.

Please have a quick look at the almost endless list of possibilities over here. These include direct loading from zip-files, gz-files, jar-files, ram drives, SMB, (s)ftp, (s)http, etc.

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

As cool examples go, here is one to end with:

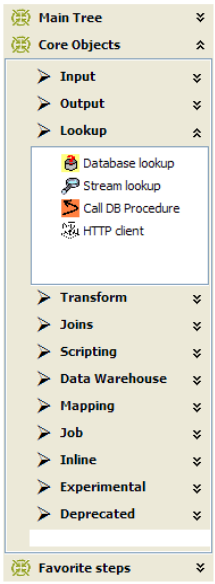



As you can see, you can use a wild-card to directly select files inside of a zip file.

Apache VFS support was implemented in all steps and job entries that are part of the Pentaho Data Integration suite.

## 2.3. Re-design of the left tree

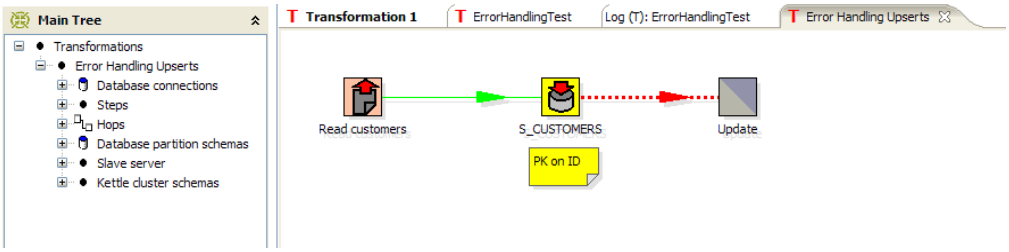
The number of job entries and steps keeps growing with every release. Also, more and more people use their own set of plugins and this only increased the number of items in the "Core steps" tree.



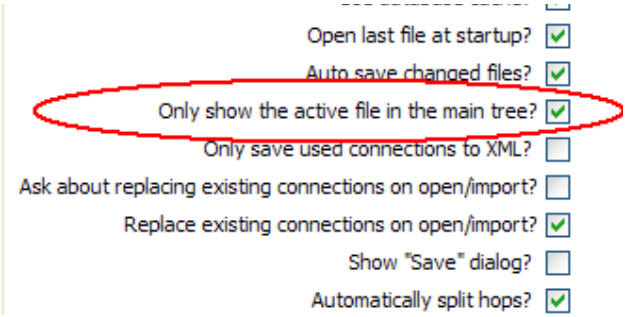

Because of that we looked at an alternative way of displaying these steps. Given the fact that you always only need one step at a time, we went with what is called an "Expand Bar". You can see an example of this on the right.

The expand bar items only expand one at a time allowing for a less cluttered GUI and easier selection from the range of options.

We also slightly changed the behavior of the "Main tree" on the top left.

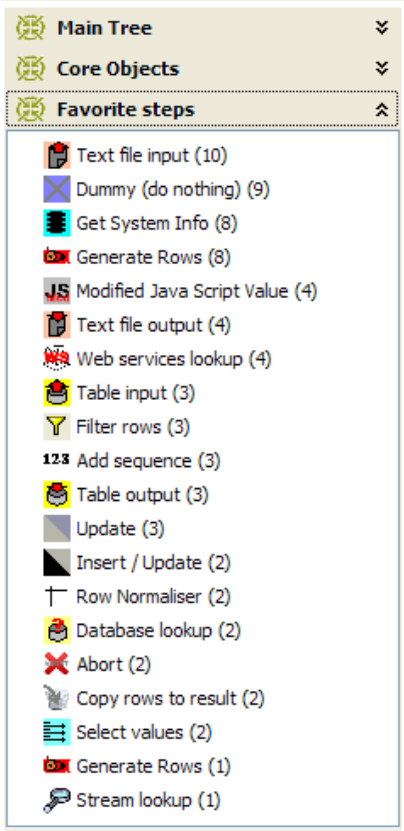



The change here is that only the selected transformation or job is displayed in because the tree would otherwise get unwieldy big to the point of being unusable. If for some reason you would still like to see them all, we added an option to influence this behavior:



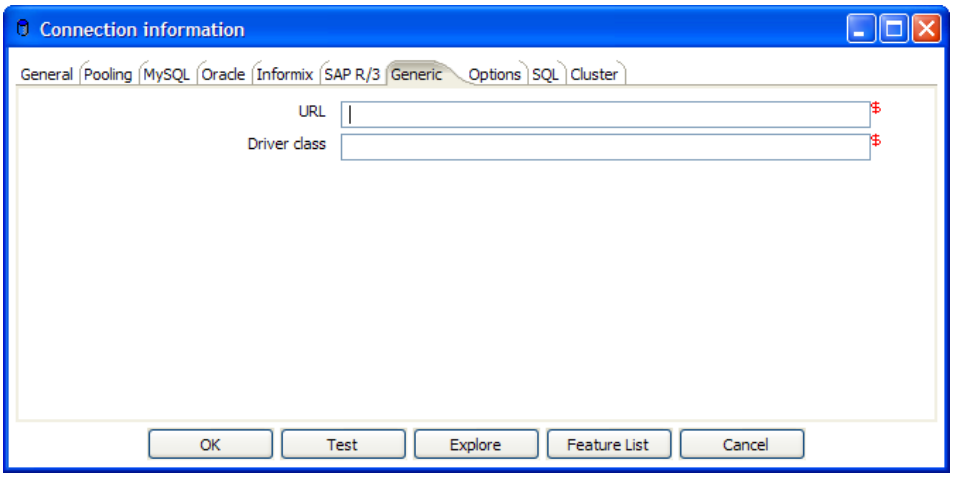

Finally, we added a "Favorite steps" section in the Expand Bar.

This is not just a history of step usage, but a ranking of the most-created steps.



## 2.4. Databases

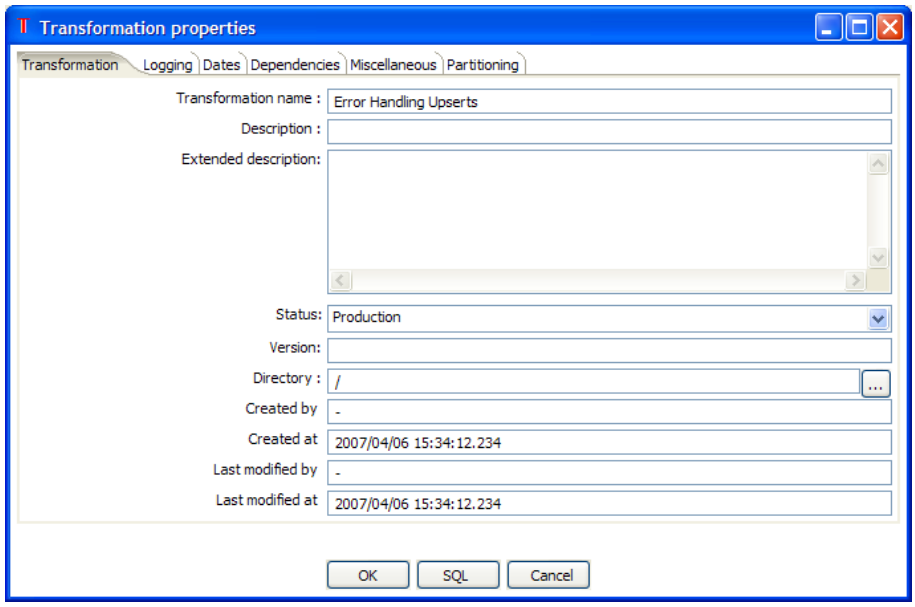
- Generic connections can now be defined using variables. That way you can source data from varying database types using the same connection in a single transformation.




- We fixed a nasty bug in the connection pooling authentication mechanism (it actually works now :-))
- Improved quoting of reserved words: when there is a start or end-quote in the tablename or schema, quoting is not done. This allows you to specify the quoting mechanism yourself. Plenty of issues where fixed with regards to schema/table SQL generation and we think that we came a lot closer to an optimal solution now.
- Added support for the Apache Derby database. The JDBC Type4 driver from version 10.2 is included.

## 2.5. Repository improvements

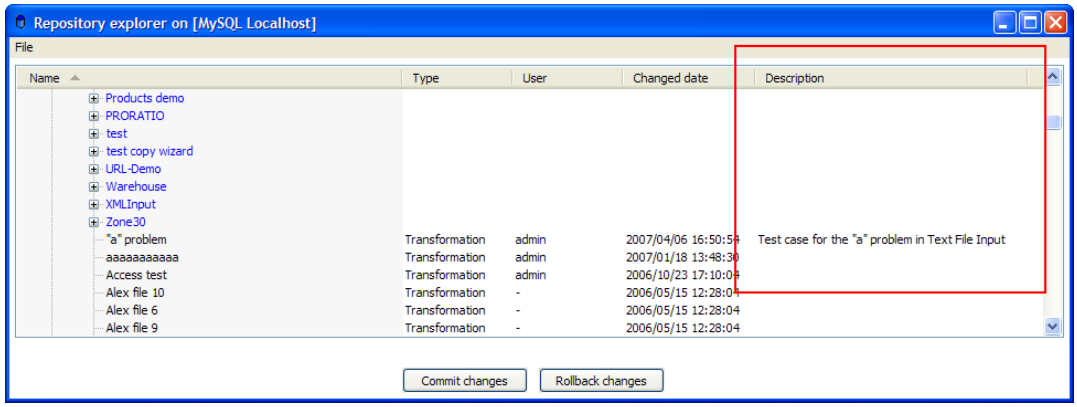
The repository got slightly changed to allow the definition of short and long descriptions of transformations and jobs.




As you can see, we added a number of fields to the transformation settings:

- Description
- Extended description
- Status (Development / Production)
- Version
- Created by / at

In the repository explorer, this is reflected by the addition of the description field. You can sort on this field as well:

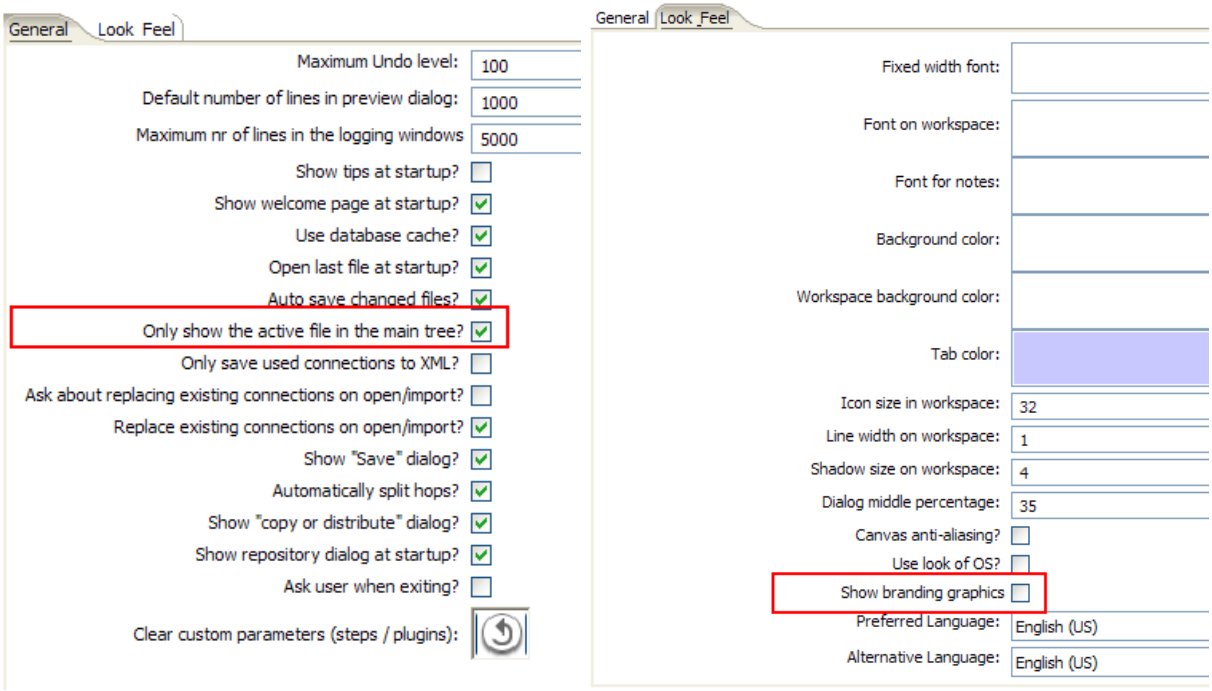


# 3. Spoon

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

## 3.1. Extra options

These are the new Spoon interface options that were added:



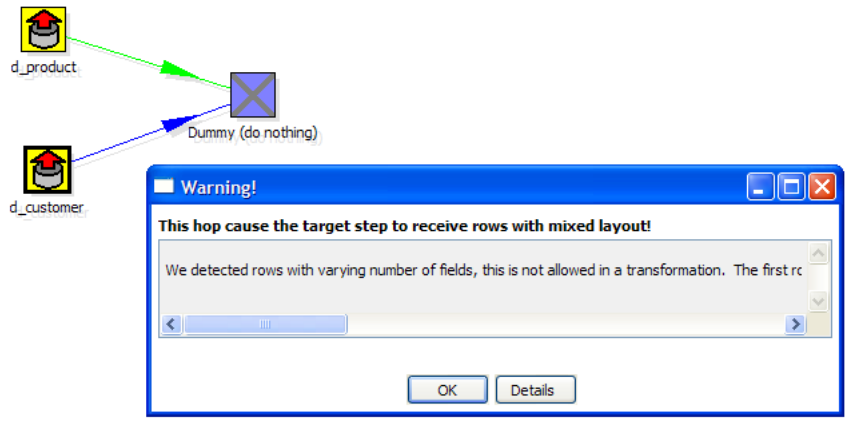

The option on the left is explained above, the “branding graphics” option is displaying some Pentaho Data Integration graphics in Spoon. (for the fans :-))

## 3.2. Mixing rows : trap detector

Mixing rows with different layout is not allowed in a transformation. However, the developers still receive many bug reports when in fact people are mixing rows of data.

This is causing steps to fail because fields can't be found where expected or the data type changes unexpectedly.

For that reason we added a “trap detector” when you by accident mix rows:




In this case the full erorr report reads:

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

# 4. Steps

## 4.1. New steps

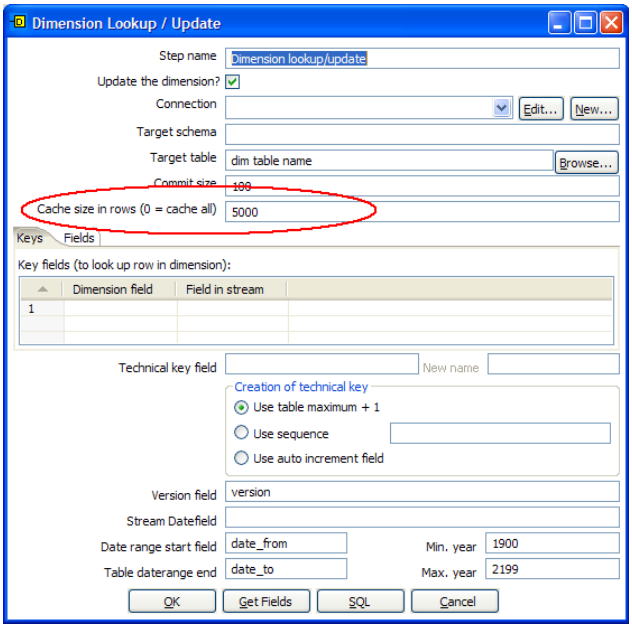
### 4.1.1. Abort step

This step can be used in combination with the new error handling functionality. It allows you to abort a transformation when one or more rows are being received.


## 4.2. Changed steps

### 4.2.1. Caching slowly changing dimensions

One of the most-requested features was the ability for the “Dimension Lookup/Update” step to cache dimension entries. This was implemented:




The caching works for both the lookup and update mode of the step.

The caching mechanism keeps the highest technical keys in memory for as long as possible because typically those keys have a higher chance of generating a cache hit.

### 4.2.2. Modified Java Script Value

This step was modified to be as compatible with the older version (Java Script Value) as possible.

### 4.2.3. Value Mapper

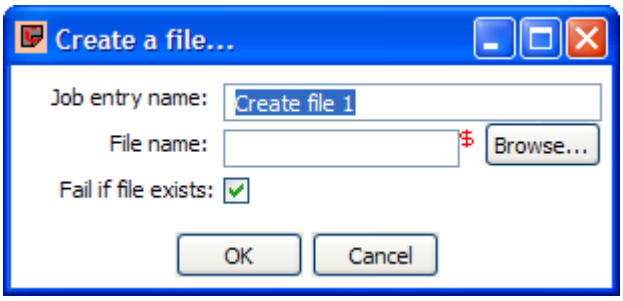
Value Mapper now also allows you to map an empty or null value to a non-empty value.

# 5. Job entries

## 5.1. New job entries

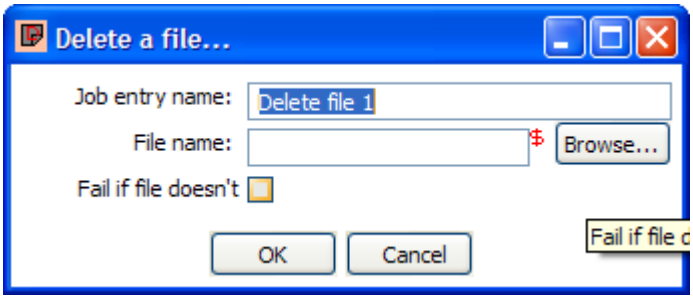
### 5.1.1. Create file

This is a simple job entry that performs a “touch” (creates an empty file).



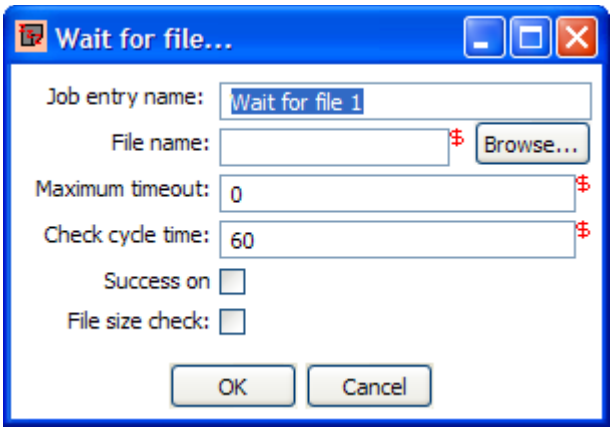

### 5.1.2. Delete file

This is a simple job entry that deletes an file



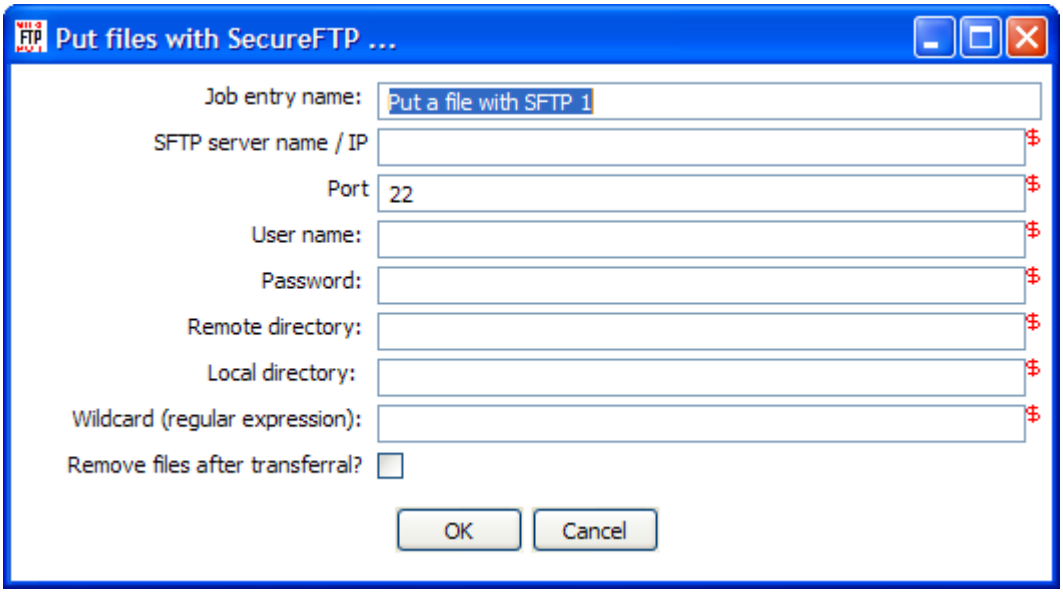

### 5.1.3. Wait for file

This job entry waits until a file appears.



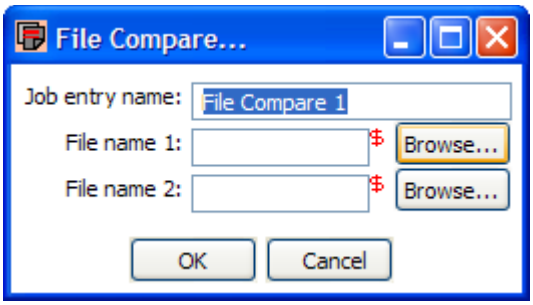
## 5.1.4. Put a file with SFTP

A job entry to allow you to put files on a secure FTP server.



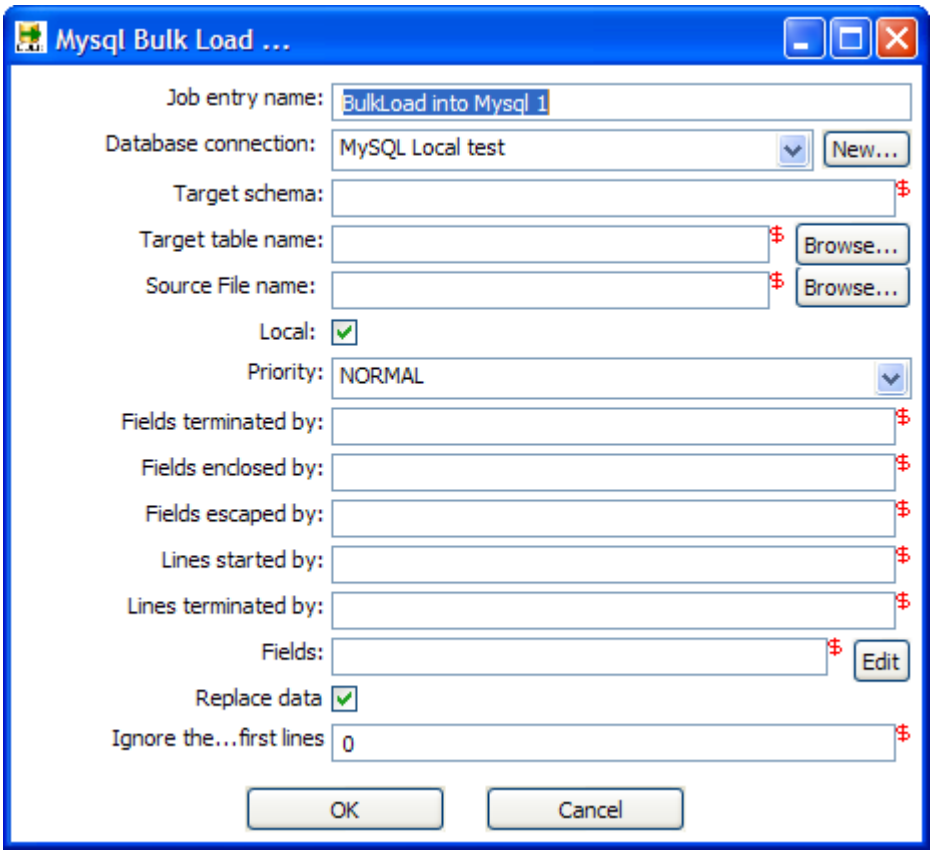

## 5.1.5. File compare

If you want to see if the contents of 2 files are identical, this is the job for you!



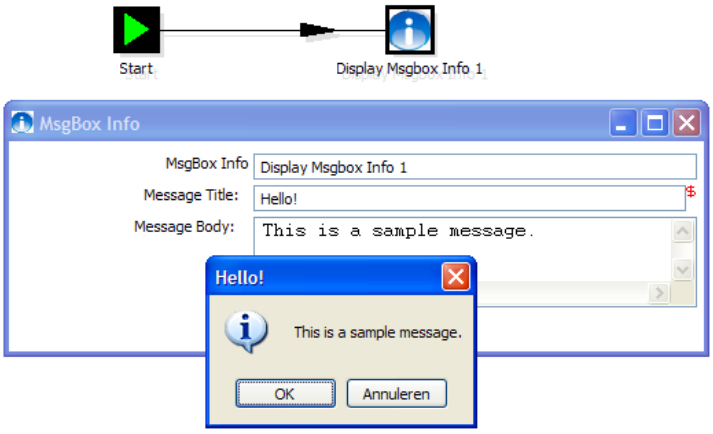
## 5.1.6. Bulk Load into MySQL

This job entry uses the MySQL specific “LOAD DATA INTO” SQL command to transfer information from a text to a database table. It has the advantage of being extremely fast.



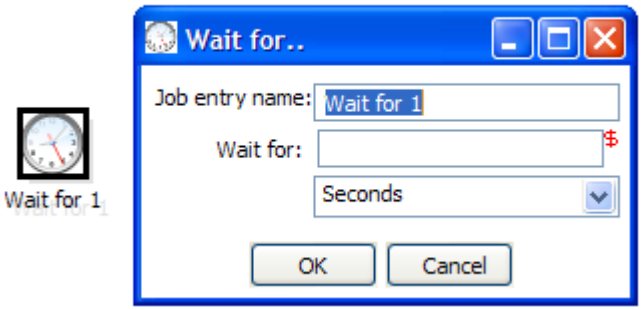

## 5.1.7. Display MsgBox Info

If you are running in the GUI, you can use this job entry to debug where you are at in the execution of a job. It is **NOT** intended nor designed for batch/runtime use.



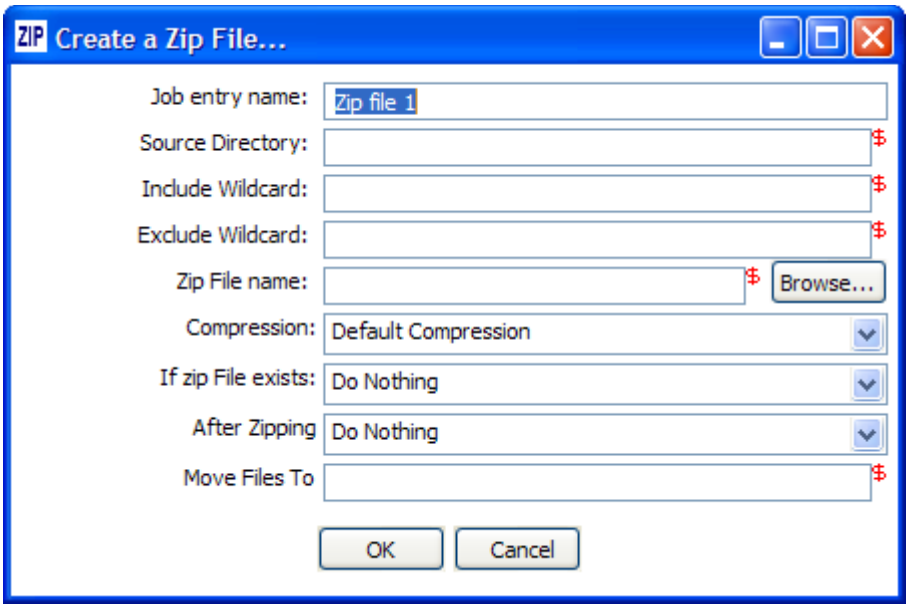
## 5.1.8. Wait for ...

If you want to put a delay in place before you retry a certain operation, you can use the Wait job entry.



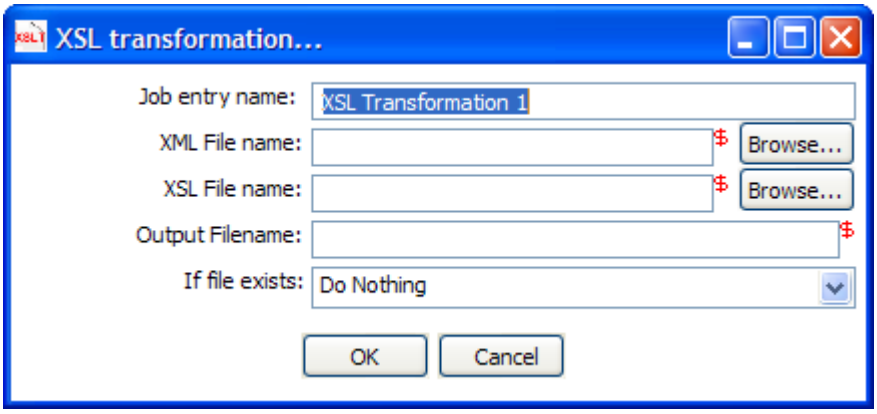

## 5.1.9. Zip file

If you want to put a number of files into a zip file, you can use this job entry.



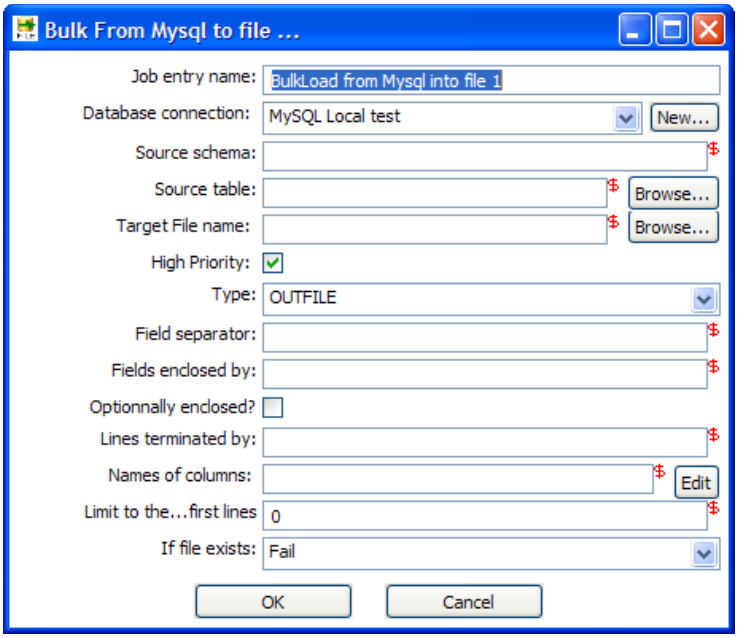
## 5.1.10. XSL Transformation

If you like to perform XSLT this is the step for you.



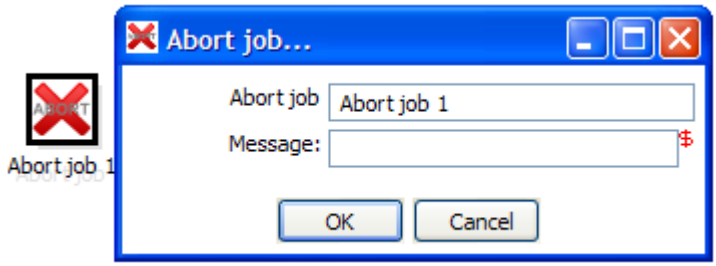

## 5.1.11. Bulk from MySQL to file

This performs a bulk export from a MySQL table to a flat CSV file.



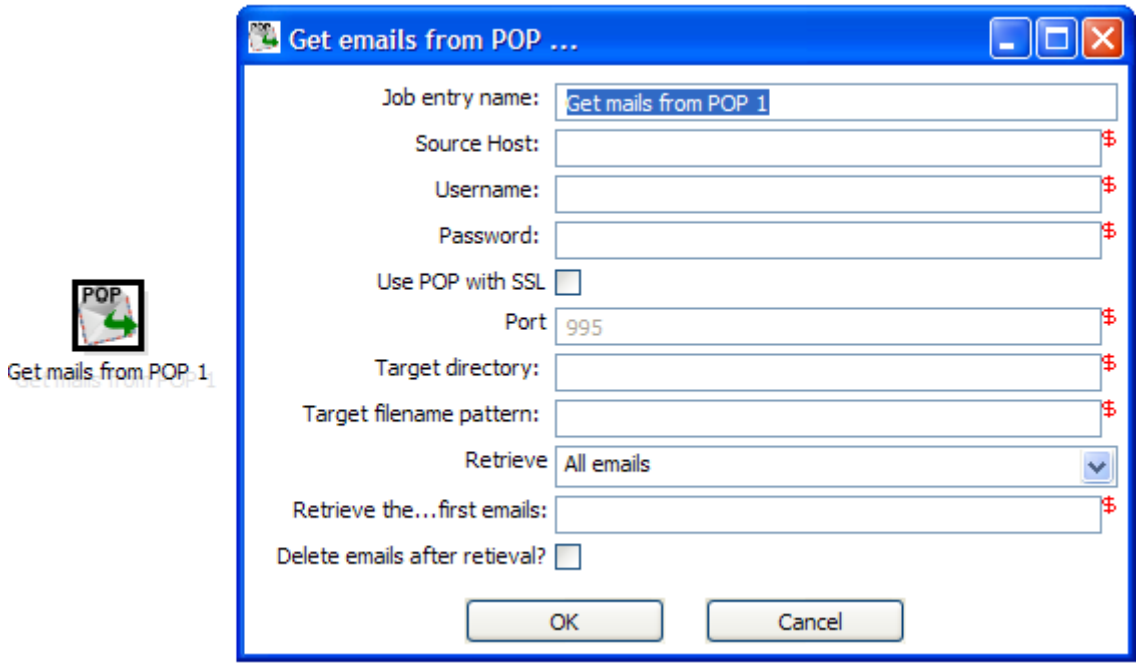
## 5.1.12. Abort job

If you want to have your job abort execution with an error, you can use this job entry.



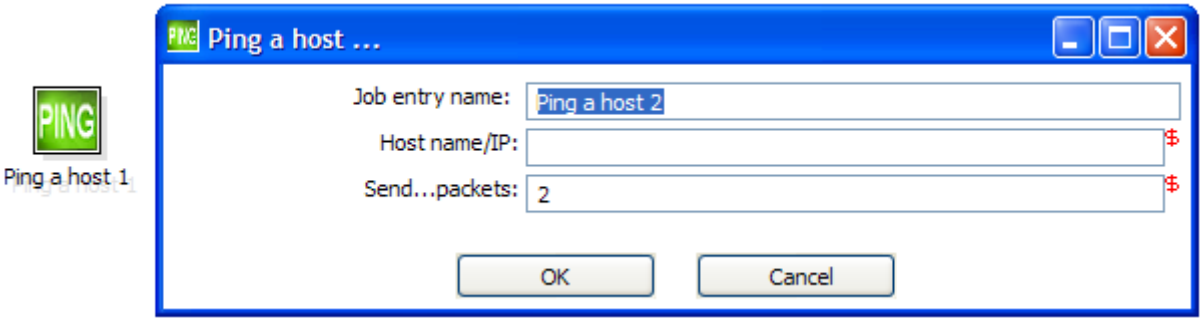

## 5.1.13. Get mails from POP

This step automatically retrieves e-Mail attachments from a POP server.



## 5.1.14. Ping a host

With this job entry you can check if a server responds to a ping (ICMP request).



# 6. Source code improvements

## 6.1. A few extra lines of code

Version 2.1.4 contains 160,000 lines of code.

Version 2.2.2 contains 177,450 lines of code, an increase of 17,450 lines.

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

Version 2.4.0 contains 256,030 lines of code, an increase of 42,541 lines.

Version 2.5.0 contains 292,241 lines of code, an increase of 36,211 lines.

## 6.2. Core committers

| ID | e-Mail | Name | Country |
|---|---|---|---|
| sboden | Svenboden (at) hotmail.com | Sven Boden | B |
| berarma | Bernardo (at) tsolucio.com | Arlandis Bernardo | ES |
| jbleuel | Jens (at) bleuel.com | Jens Bleuel | D |
| Sven.thiergen | s.thiergen (a) itcampus.de | Sven Thiergen | D |
| shassan2 | Sahass78 (a) yahoo.fr | Samatar Hassan | F |
| molm | manfred.olm (a) free.fr | Manfred Sherlock Olm | F |
| m.stoedtler | m.stoedtler (a) alea.de | M. Stoedtler | D |
| hdupre | henri.dupre (a) gmail.com | Henri Dupre | F |
| mcasters | mcasters (at) pentaho.org | Matt Casters | B |

Because of time constraints and a move to a different subversion server, it was very difficult to get a detailed commit log ready in time for the release. It's also very hard to quantify the amount of work someone did as it is very clear that we appreciate all the help we get on this project.

That being said, I would like to offer special thanks to Sven and Samatar for their never ending stream of bug fixes and good solid code and for all the work they did out on the forum.

Many thanks also go to the Pentaho team for supplying us with a new home for our Subversion server so quickly after the JF server crash of March 29th. Especially Brian Hagan went beyond the call of duty to get this done on such short notice.

## 6.3. Commit stats

In total we had around 800 revisions of the source code (commits) for in total 2554 file operations. (modification, addition, deletion)

For a detailed log of the revisions, please see the file "`Revisions-240-250.txt`" in directory docs/English.

## 6.4. Bug reporters

***Thank you all! Without you, it would be impossible to get PDI as stable as it is today.***

## 6.5. Feature requesters

***Thank you all for the many good suggestions! Sometimes the simple things have a great impact on the software. Good ideas really make the difference.***