KETTLE 2.3.2 FEATURE LIST

Document version: 2006/10/03

Supported database	**Read**	**Write**	**Repository**	**Database Explorer**	**ODBC**	**JDBC**	**OCI**	**JNDI**
MySQL	√	√	√	√	√	√		√
Oracle	√	√	√	√	√	√	√	√
AS/400	√	√	√	√	√	√		√
MS Access	√	√	√	√	√	√		√
MS SQL Server	√	√	√	√	√	√		√
DB2	√	√	√	√	√	√		√
PostgreSQL	√	√	√	√	√	√		√
Intersystems Caché	√	√	√	√	√	√		√
Informix	√	√	√	√	√	√		√
Sybase	√	√	√	√	√	√		√
Gupta SQL Base	√	√		√	√	√		√
dBase III, IV or 5	√	√		√	√	√		√
Firebird SQL	√	√	√	√	√	√		√
Hypersonic	√	√	√	√	√	√		√
MaxDB (SAP DB)	√	√	√	√	√	√		√
CA Ingres	√	√	√	√	√	√		√
SAP R/3 System	√				√	√		√
Generic database access	√	√		√	√	√		√
Borland Interbase	√	√	√	√	√	√		√
ExtenDB	√	√	√	√	√	√		√
TeraData	√	√		√	√	√		√
Oracle RDB	√	√		√	√	√		√
H2	√	√	√	√	√	√		√

Supported Platforms	**GUI**	**Runtime**
Windows: 95, 98, ME, NT, 2000, XP, 2003, Vista	√	√
Linux	√	√
Solaris	√	√
Apple OSX: PowerPC and Intel	√	√
HP-UX	√	√
AIX	√	√
AS/400 (iSeries)		√
FreeBSD		√

The runtime is also available on any system running Java Runtime Environment version 1.4.2 or higher

Text file reading
Read: single or multiple files, static or dynamic filename definition
Read: CSV files with optional separator, enclosure, escapes, etc.
Read: Support for zipped files
Read: Fixed record length files
Read: automatic scanning of source file(s)
Read: paged layout documents (printed reports)
Read: support for document encoding specification (Unicode, etc)
Read: advanced error handling with recovery system
Read: filtering out of unwanted records

Text file writing
Write: field selection
Write: output format selection
Write: Append file
Write: Split file into parts based on the number of rows
Write: separator and enclosure can be specified
Write: support for zipped files

Metadata
All Kettle actions and functions are defined as metadata (100%)
All Kettle metadata can be expressed in XML
All Kettle metadata can be expressed in the Kettle database repository
All Kettle metadata can be edited using a GUI (Spoon, Chef, Dialogs, ...)
The Kettle metadata is always in a readable format: open access
Metadata of jobs, transformations, database connections, etc. can be searched

Kettle Repository
Create a Kettle repository: relational database tables with reference content
Upgrade a Kettle repository from a previous version or re-populate the reference content
Create and administer repository users & passwords
Create, Delete, Edit or rename user profiles
Grant rights to users using profiles (read-only, administrator, normal user, …)
Store, retrieve, rename, delete transformations in the repository
Store, retrieve, rename, delete jobs in the repository
Create, Edit, Store, retrieve, delete database connections in the repository
Create, Delete or rename directories and subdirectories in the repository
Allow jobs and transformations to be moved from one directory to another (drag & drop)
Export all repository objects to an XML file (for backup)
Import all repository objects from XML file to a destination directory (for recovery)
"Export all transformations" functionality
"Export all jobs" functionality

Transformation functionality
Transformation: a collection of steps performing specialized tasks (see below)
Two or more steps pass rows of data from one step to another
Work with rows of fields containg data
Merge streams of rows from different sources
Split streams of rows to multiple targets (for example: write to database and text file at the same time)
Origin tracking of fields. (where is field X coming from)
Unlimited number of in-between steps (depending on available memory and CPU)
Unlimited number of processed rows
No creation of temporary files except for sorting and joining.
Plugin-system allowing any java programmer to write a transformation step
Data types: - double precision floating point(64 bit) - long integer (64 bit) - date (millisecond precision) - string (unlimited length) - boolean (true/false) - big number (any precision or length) - binary objects (Images, sounds, etc)
Even distribution of source rows to multiple target steps. (split load over multiple databases)
Copy of source rows to multiple target steps. (send the same data to multiple databases)
Allow a step in the transformation to be started more then once in parallel.

Available steps
DB: Parameterized read from database
DB: Write to database table (using batch loads), dynamic target table support, partitioning support
DB: Perform lookups in table
DB: Database join using freehand SQL
DB: Insert / Update database table function
DB: Update database table function
DB: Call database procedure/function with parameters
DB: Delete rows in a table
Lookup values in memory using data from any source
Get information from the environment (time, date-ranges, arguments, …)
Generate one or more rows
Native read from Xbase files (dBase III, IV, Foxpro, …)
Native read and write from Microsoft Excel files, read data from separate sheets, range selection, …, including calculations
Select fields in rows from previous steps, delete fields, change name/type of fields (meta-data)
Filter rows based on conditions, using graphical condition editor
Sort rows based on a combination of fields (ascending and descending)
Add sequences using transformation wide counter or using Database sequences
Join multiple streams of rows in a cartesian product with optional conditions.
Execute any JavaScript on the fields in the rows
Normalize values in crosstabs (for example from Excel forecasts, etc)
Remove double rows (unique function) by looking at specified fields for equality, includes support for case sensitivity
Grouping function allowing the calculation of sums, counts, … includes the option to have the original rows next to the aggregation values
High-speed calculator functionality allowing pre-defined much used functions to be executed such as percent calculation, rounding, etc.
Field splitting for complex parameterized data (A=100,B=200,D=400, etc)
Execute arbitrary parameterized SQL commands using rows for input.
Create mappings (sub-transformations) to create re-usable objects

Read content from XML files
De-normalize values by doing key-value lookups
Merge rows step to detect changes between 2 data streams
Add constant values
Flatten consecutive row values into different field values
Value mapper allowing quick data conversion/corrections to take place.

Spoon GUI
GUI designer for transformation using drag&drop and easy to use dialogs
Copy / Paste of steps
(multiple) undo redo of changes
Determine input fields for a step showing the origin
Determine output fields for a step showing the origin
Keep a list of the last opened files.
Change colors, background colors and fonts used in Spoon
Allow hops between steps to be split
Allow steps to be detached from hops between steps.
"Create database connection" wizard
"Copy database table" wizard
Run transformations
Preview the rows passing any step in a transformation
End-to-end checking of transformation (or selected steps)
Allow database operations to be cancelled (for example long running queries)
Progress dialogs for time-consuming operations
Database Impact analyses of a transformation (Read, Write, Update, …)
Persistent caching of SQL results
Read/Write transformations from/to XML files
Copy/Paste transformations in XML from/to the clipboard
Read/Write transformations from/to one or more Kettle repositories
Connect to a Kettle repository
Explore a Kettle repository
Share/manage database connections in a central Kettle repository
Database: explore a database using the database connections
Database: execute SQL using the SQL Editor
Database: create SQL for table, index or sequence creation or modification
Define new Kettle or environment variables

Create image of shown transformation (BMP format) to clipboard
Search the meta-data of the transformation for content
Ask values of undefined command-line arguments and variables before running
Show the history of the transformation execution in the log history window based on the logging table