



Knocker main application User manual

Author: Jaroslav Tykal
Application: Knocker.exe



Content:

1	START APPLICATION	3
1.1	CONNECTION TO DATABASE	3
1.2	MODULE DEFINITION	3
2	CONNECTION TO DATABASE	4
2.1	MANAGE DATABASE CONNECTIONS	4
2.1.1	How to show this dialog:	4
2.2	ADD A NEW CONNECTION	5
2.2.1	Items:	5
2.2.2	Sample ODBC connection for MySQL	6
2.3	REMOVE CONNECTIONS	6
2.4	SET OTHER CONNECTION AS "WORK"	7
2.5	ADD SUPPORT FOR UNKNOWN ODBC DRIVER	7
2.5.1	Changes in configuration file	7
3	MANAGE MODULES	10
3.1	MANAGE MODULES (METHODS)	10
3.2	ADD MODULE	10
3.2.1	Items	11
4	MAIN APPLICATION WINDOW	12
4.1	MAIN MENU	12
4.1.1	Main	12
4.1.2	Version	13
4.1.3	Methods	14
4.2	CONTEXT MENU	14
4.3	SIMPLE TRANSFORMATIONS	14
4.4	PROCEDURES	14
4.4.1	Create new version of data	14
4.4.2	Drop version	15
4.4.3	Apply simple transformations	15
4.4.4	Apply simple transformations on other version	16
4.4.5	Run data mining method	16
4.4.6	Save session	17
4.4.7	Create new session	17
5	SOLUTION OF POSSIBLE PROBLEMS	18
5.1	CAN NOT CONNECT TO MS ACCESS DATABASE (SQL ERROR)	18

1 Start application

For a correct running of the application is necessary:

- a) to define a connection to database
- b) to define modules which work with data and visualize data mining methods

1.1 Connection to database

It is necessary to define a connection to database at the first application run. This database will be used as data storage. The database connection is always called „work connection“. Data storage will contain all versions of all sessions. Each version contains either a copy of existing table or data imported from CSV file or modified data from existing version created by existing modules.

Application allows user to define more than one database connection. But just one can be the „work connection“. More connections are useful when user requests data from more sources (e.g. from tables in MS Access, MS SQL, Oracle, MySQL)

Application checks an existence and validity of „work connection“ during the start initializations. If there is no valid „work connection“ (application can't connect using this connection), the application will show dialog to manage connections.

Detail information is included in chapter [„Manage database connections“](#)

1.2 Module definition

We can use two different types of modules.

- Simple transformations - data transformations. It is useful for preparing of data. E.g.: normalization, null values replacing, splitting data from one table into more tables, etc.
- Normal transformations (we call them „methods“) - these modules visualize the work of selected Data mining methods. E.g.: Neural networks, Market basket analysis, Decision trees, Kohonen maps.-

The transformations of the first type are additive. More transformations of this type can be reapplied to similar data.

The transformations of the second type haven't got this facility (they are not additive). Methods implemented visualization of algorithm instead.

2 Connection to database

All database connections use ODBC driver. ODBC driver should be installed on computer for each database you intend to use.

Application was tested with these ODBC drivers:

- Microsoft ODBC driver for Oracle
- Oracle ODBC driver
- MySQL ODBC 3.51 driver
- Microsoft ODBC driver for Access
- Microsoft SQL Server 2000

It is possible to enlarge this set of databases by modification of configuration file. More info is in chapter [„Add new type of ODBC driver“](#).

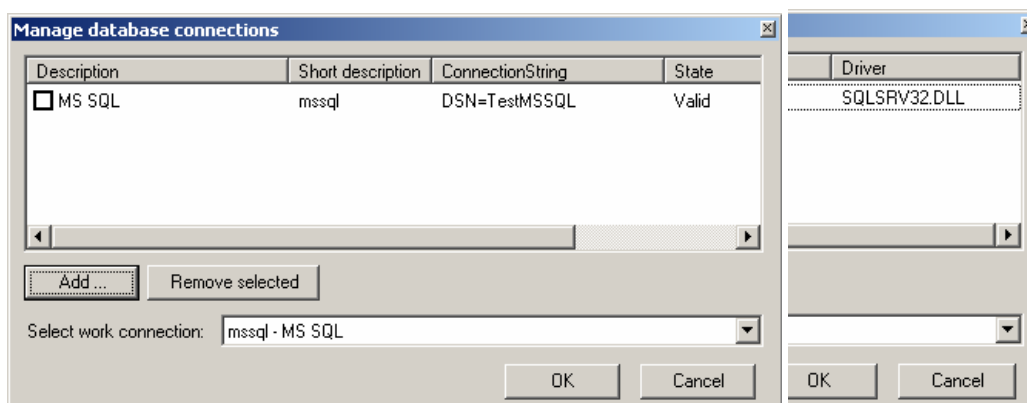
Information about defined connection is stored in configuration file *connections.xml*.

Warning

Defining appropriate database as the „work connection“ may affect time of processing.

We recommend using of MS Access just as a data source – do not use it as database for „work connection“.

2.1 Manage database connections



2.1.1 How to show this dialog:

- This dialog will be shown automatically if no valid work connection exists
- main application window menu: Main-> Database connections ...

Each database connection contains:

- Description – description for user
- Short description – this description is shown whenever user displays a connection or a table/connection (e.g. in combo box for select tables).
- Connection string – connection string which is used to connect to database.
- State – current state of database connection. This item can reach one of the following values:
 - o Valid – connection is correct
 - o SQL Error – connect is correct, but SQL-query for obtaining a list of all table names failed. This error is typical for wrong permission settings on database system tables.
 - o Broken – it is impossible to connect to database.

- Driver – When the connection to database is correct, the Driver for this ODBC connection is shown. Otherwise text „Broken” is shown.

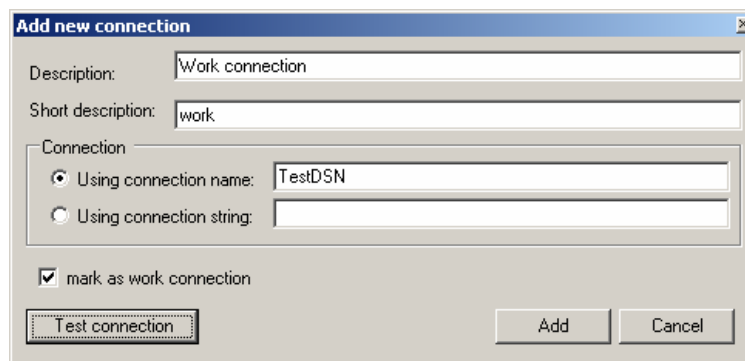
Capabilities:

- Add new connection
- Remove connections
- Set connection as the „work connection”

2.2 Add a new connection

How to show this dialog:

- click on „Add ...” button in Manage database connection dialog.



2.2.1 Items:

- Description – description for user.
- Short description – this name is shown whenever user selects a connection or table/connection (e.g. in combo box for select tables).
- Connection:
 - Using connection name – in this case a connection is defined in user or system DSN (Control panels -> Administrative tools -> Data Sources (ODBC)). See also [Sample ODBC connection for MySQL](#).
 - Using connection string – in this case user has to enter a valid connection string to connect to database.
- Mark as work connection – if this choice is checked then connection is set as the „work connection”. Connection previously marked as the „work connection” remains in definition of connections, but till that moment it will not be marked as „work connection”.

Click on „Test connection” button to detect whether the connection is valid. This test takes effect just in case that Connection name (possibly Connection string) is not empty

Results of this test might be:

- Connect to specified database is OK
- This connection do not support show list of all tables
 - This result may happen in case of incorrect definition of configuration file (*sqlStatements.xml*) or if permission setting for system database tables failed.
- Unknown ODBC driver. Please, add relevant information about driver into configuration file.
 - This result is returned when user selects a data source which is unknown for application. See also [Add support for unknown ODBC driver](#).

A new connection insert is confirmed by clicking on “Add” button.

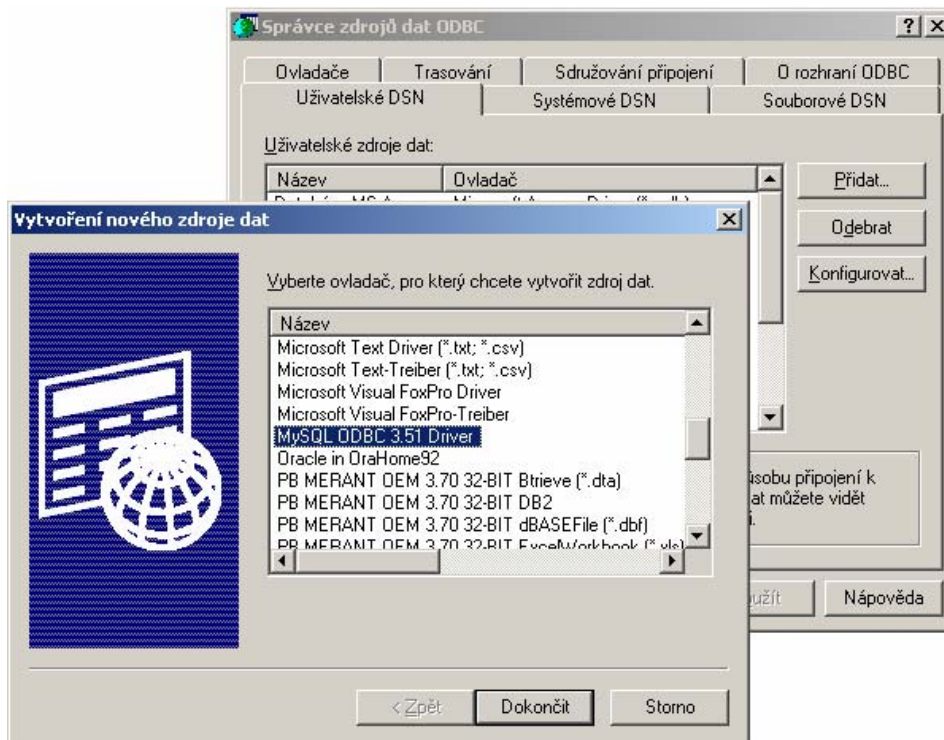
Note:

User can add an invalid database connection. The connection must be valid at application start or it cannot be used.

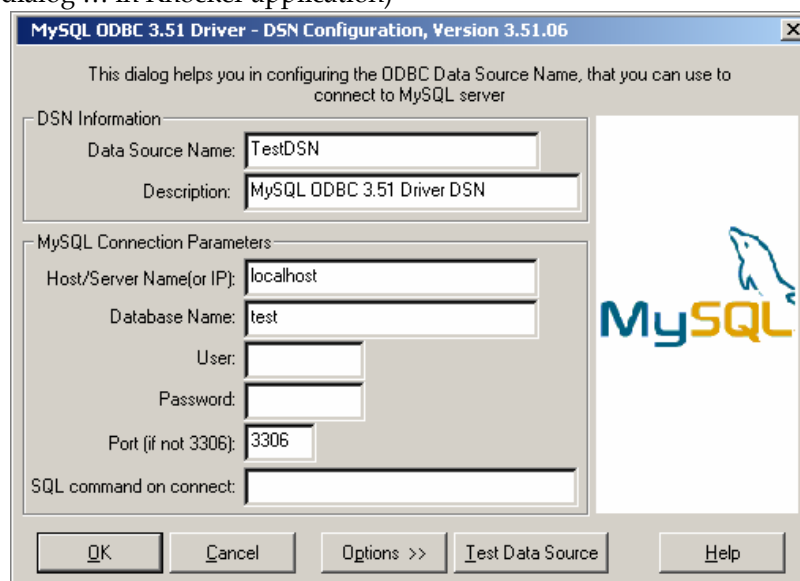
2.2.2 Sample ODBC connection for MySQL

In Windows: Open Start -> Control panels -> Administrative tools -> Data Sources (ODBC).

1. Open User DSN (Uživatelské DSN)
2. Click on „Add ...“ button (Přidat ...)
3. Select MySQL ODBC 3.51 (or similar name – depends on installed ODBC driver)
4. Click on “ Finish” (Dokončit)



5. enter Data Source Name (this Data Source Name is used as the connection name in Add new connection dialog ... in Klocker application)

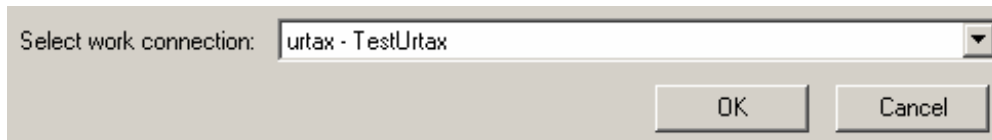


2.3 Remove connections

Each connection has a check box in Manage database connection dialog. All checked connections will be removed after pushing the „Remove selected“ button.

2.4 Set other connection as “work”

There is a combo box with all defined connections in Manage database connection dialog.



Choose one of the connections and confirm by clicking on “OK” button.

2.5 Add support for unknown ODBC driver

ODBC interface may differ for some types of queries or databases. E.g. list of all tables; all databases don't use the same data types, etc.

That's why it is necessary to make mapping between SQL statements and corresponding database. This mapping is in a configuration file *sqlStatements.xml*.

Application can find out only driver name from the database connection. In practice, more databases can be managed by one driver (DLL library).

2.5.1 Changes in configuration file

User must do these steps after insert of unknown (for application) driver

1. obtain driver name - DLL library with new ODBC driver
2. modify configuration file *sqlStatements.xml* with SQL statements and data types

1. obtaining a driver name

In Windows: Start -> Controls panel -> Administrative Tools -> Data source (ODBC) -> Drivers. Each driver has information about relevant file with Driver.

2. modification of the configuration file

Each record in element “database” contains a definition of one database and one or more drivers.

Each database object may refer to more than one element “database” in the configuration file. This usually happens when a driver is interface for more databases. Application finds the first database name in which the driver corresponds. Then tries to execute “showTables” statement and checks if the statement fails or not.

Example:

```
<database name="Oracle">
  <description>ODBC for Oracle</description>
  <dbdriver>MSORCL32.DLL</dbdriver>
  <dbdriver>SQORA32.DLL</dbdriver>
  <conversions>
    <columnInt>integer</columnInt>
    <columnDouble>double precision</columnDouble>
    <columnString>varchar2(255)</columnString>
    <columnLongString>varchar2(4000)</columnLongString>
  </conversions>
  <sql>
    <showTables>SELECT table_name FROM cat</showTables>
    <alterTableModify>ALTER TABLE :TableName MODIFY :ColumnName
:ColumnTypeNew</alterTableModify>
    <alterTableAdd>ALTER TABLE :TableName ADD :ColumnName
:ColumnType</alterTableAdd>
    <alterTableRemove>ALTER TABLE :TableName DROP COLUMN
:ColumnName</alterTableRemove>
```

```
<alterTableRename>ALTER TABLE :TableName RENAME COLUMN
:ColumnNameOld TO :ColumnNameNew</alterTableRename>
<dropTable>DROP TABLE :TableName</dropTable>
</sql>
</database>
```

Element `<database>`:

- It contains definition of SQL statements for group of ODBC drivers.
- Attribute name is optional and it has been made for better database identification

Element `<description>`: optional, at most one

- Long description of this record.

Element `<dbdriver>`: mandatory, at least one

- Name of driver – DLL library with ODBC driver.

Element `<conversions>`: mandatory, exactly one

- It contains other elements for processing of conversion between database types and „simple types“

Element `<columnInt>`: mandatory, exactly one

- Database type corresponding to simple type INT – an integer

Element `<columnDouble>`: mandatory, exactly one

- Database type corresponding to simple type DOUBLE – a number with decimal point

Element `<columnString>`: mandatory, exactly one

- Database type corresponding to simple type STRING – text, up to 255 characters length

Element `<columnLongString>`: mandatory, exactly one

- Database type corresponding to simple type STRING – text, up to 4000 characters length

Element `<sql>`: mandatory, exactly one

- It contains other elements with SQL statements. It is necessary to define following items:
 - show list of all tables (element `<showTables>`)
 - change database type of relevant table column (element `<alterTableModify>`)
 - add column into relevant table (element `<alterTableAdd>`)
 - remove column from relevant table (element `<alterTableRemove>`)
 - rename relevant table column (element `<alterTableRename>`)
 - drop table (element `<dropTable>`)

It is necessary to use variables for table name, column name and type. These variables will be substituted with appropriate values.

These variables has prefix ‚:‘ and their names are described near each element description.

Each of following elements are mandatory and occurs exactly once in each ‚database‘ element.

Element `<showTables>`:

- Shows list of all tables for connected user. No variables are available.

Element `<alterTableModify>`:

- SQL statement for type of column modification.

Available variables are:

- :TableName – name of the table
- :ColumnName – name of the column
- :ColumnTypeNew – new database type of the column

Element `<alterTableAdd>`:

- SQL statement for inserting column into the table.

Available variables are:

- :TableName – name of the table
- :ColumnName – name of the new column
- :ColumnType – database type for the new column

Element `<alterTableRemove>`:

- SQL statement for removing one of existing columns from the table.
Available variables are:
 - `:TableName` - name of the table
 - `:ColumnName` - name of the column to remove

Element `<alterTableRename>`:

- SQL statement for renaming of the column.
Available variables are:
 - `:TableName` - name of the table
 - `:ColumnNameOld` - old name of the column
 - `:ColumnNameNew` - new name for a column

Element `<dropTable>`:

- SQL statement for dropping of the table.
Available variables are:
 - `:TableName` - name of the table

If there are multiple drivers belonging to one database and these drivers have the same set of SQL statements, then they can be placed in one shared "Database" element.

Drivers for different databases should always be placed in separate "Database" elements although the set of SQL statements for all the drivers is identical. This rule keeps the configuration file well-arranged and readable for human.

3 Manage modules

Both types of modules (simple transformations and methods) can be loaded to application in the same way - [Manage modules dialog](#).

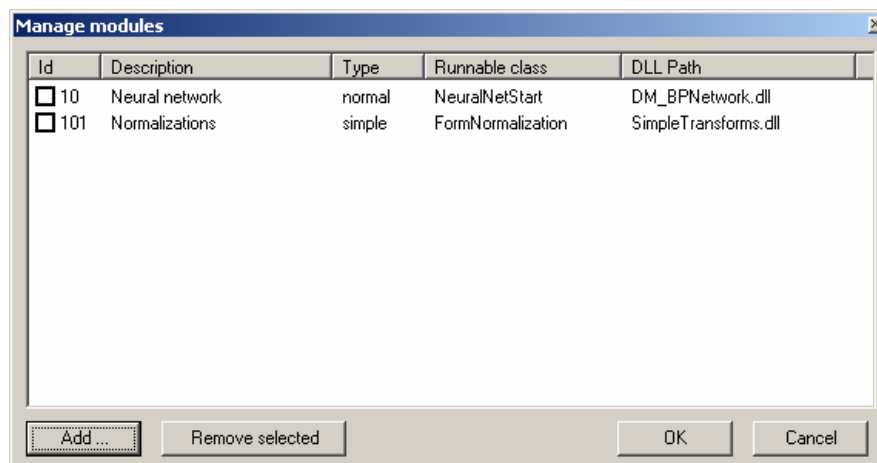
3.1 Manage modules (methods)

How can user show this dialog?

From main windows menu: Methods -> Methods ...

Information about each module:

- ID - helps differentiation among methods. Each method must have unique ID.
- Description - This text will represent this method in lists of all methods.
- Type - type of method
 - simple - simple transformations for simple data transform
 - normal - more complex methods which support visualization
- Class to run - name of the class which is used to run module
- DLL Path - location of DLL library with method

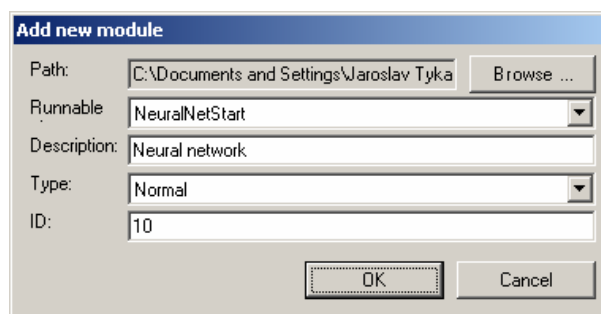


Capabilities:

- Add method
- Remove methods

3.2 Add module

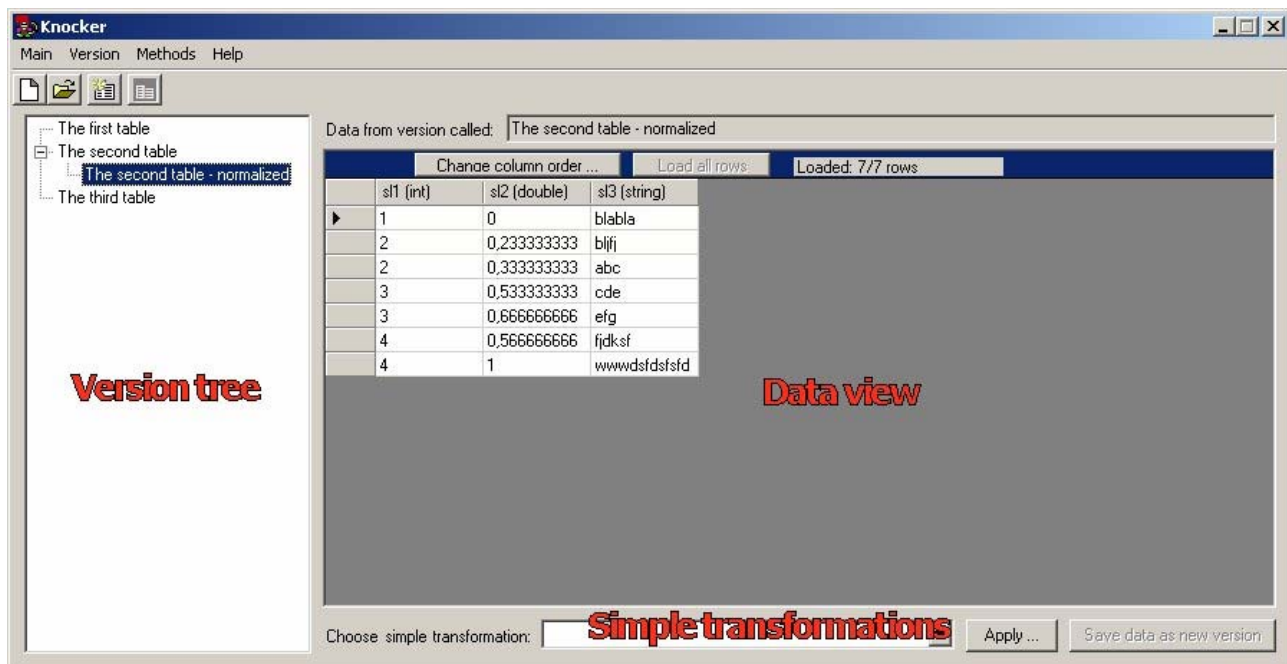
In Manage module dialog, click on "Add ..." button.



3.2.1 Items

- Path - at the beginning it is necessary to select a DLL containing desired module. User can find the DLL file by browsing - „Browse ...“button.
 - Class to run - when program reads selected DLL library, a list of all classes accessible in DLL is shown. As a next step, user selects class, which is used as a starting point of the method. (You can find name of this class in documentation of each module).
 - Description - This text will appear in main application menu (for „normal“ data transformations) or in main application combo box (for „simple“ data transformation).
 - Type - „Simple“ (transformations without visualization support) or „normal“.
 - ID - should be unique for each module.
-

4 Main application window





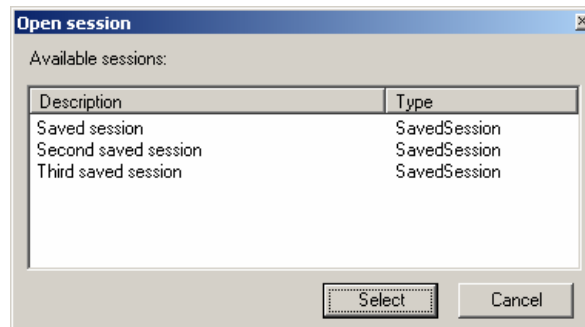
Capabilities:

- Version tree.
 - reading data into version tree
 - dropping version (and all its subversions) from version tree
 - showing data for selected version
 - showing basic information about selected version
- Data view
 - user can choose which columns are displayed
 - user can choose column order in data grid
 - user can load all data in data grid; this action may take allocate huge part of memory
- Running method of type „normal“.
- Switching to Edit mode – user can repeatedly apply simple transformation on selected version. Result may be saved as derived version into Version tree.
- Managing database connections.
- Managing methods.

4.1 Main Menu

4.1.1 Main

- **New** 
This command creates a new session. If another previously opened session is active, user will be asked whether the active session should be saved or not.
- **Open ...** 
This command opens a dialog to select session from list of existing sessions. If another previously opened session is active, user will be asked whether the active session should be saved or not.



Type of session can contain one of these values:

- o SavedSession – session is saved in database
- o NewSession – session is not saved in database
- o OpenSavedSession – another user has probably opened this session

- **Close**

This command closes current session and opens a new session. If another previously opened session is active, user will be asked whether the active session should be saved or not

- **Database connections ...**

This command shows a dialog to administrate database connections.

- **Exit**

This command closes application

4.1.2 Version

- **Add new version ...** 

This command opens a dialog to load data from a database table or from a file in CSV format.

- **Drop selected version**

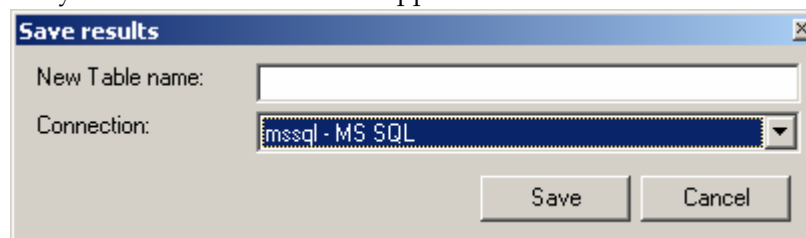
This command deletes selected version.

- **More info ...**

This command shows a dialog with more information about selected version.

- **Save version as permanent table ...**

This command saves selected version as a permanent table into the working database. This table doesn't belong to any session and will not be dropped.



New table name must not be empty, the first character is in {'a', ..., 'z', 'A', ...'Z'} and other characters are in {'a' - 'z', 'A' - 'Z', '0' - '1', '_'}. Maximal length of this name is 20 characters.

- **Save version as CSV ...**

This command exports data from selected version into a file in CSV format. File uses UTF8 encoding. Data are in format described in [Create new version of data](#).

- **Copy transformation**

If a version is created using simple transformations then these transformations may be saved into “clipboard”.

- **Apply transformation**

If a simple transformation can be applied to the selected version then this command executes it. Then it creates a new version derived from the selected one.

4.1.3 Methods

- **Methods ...**

This command shows a dialog to add/remove modules.

- ...

Each “normal” module is shown as an item of this menu. This item is responsible for corresponding module start.

4.2 Context menu

Context menu for the Tree View Control has the same menu (except Add new version ...) items as you can find in Main Menu: menu Version.

4.3 Simple transformations

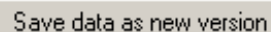
- Selection of all simple transformations



- Push the button to apply selected transformation



- Save modified data as a derived version




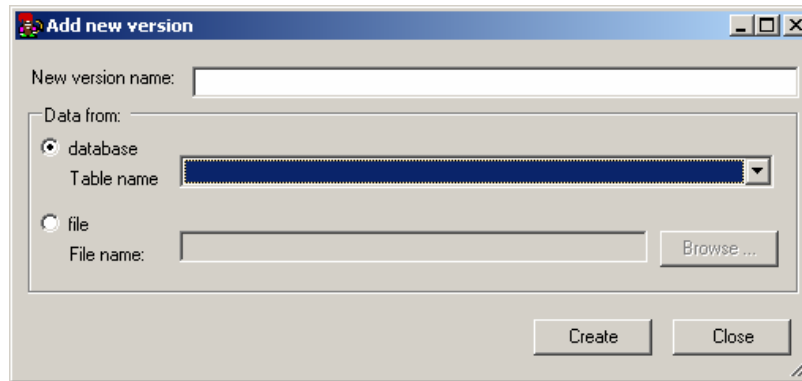
4.4 Procedures

4.4.1 Create new version of data

New version can be created from the following types of data:

- existing database table (from defined connection)
- file in CSV format (format described below)

1. Run dialog for a new version creation. This can be done by clicking on  or by choosing an item Version -> Add new version ... from menu
-



2. Enter a new version name. This name will be shown in the version tree.
3. If our version should be created from table then select database and table from the combo box. There are only names of the tables which do not contain any space in their name.
If version should be created from a CSV file then user finds the file using “Browse ...” button or writes file name directly into the text box.
4. Confirm by pressing “ Create” button.

Note:

Format of CSV file:

- the first line contains a name for each column
- the second line contains simple type for each column
 - o Simple type is one of values: INT (integer), DOUBLE (number with decimal point), STRING (text with length up to 255 characters).
- each next line contains data
 - o empty column is interpreted as null value

Sample of CSV file:

```
Column1;Column2;Column3  
INT;DOUBLE;STRING  
1;1,5;the first line  
2;;the second line
```

4.4.2 Drop version

Prerequisites:

- version which should be dropped exists

User can drop a version in two ways:

- To click the right mouse button on the version and select Drop from context menu.
- To select the version and choose Version -> Drop selected version from Menu.

4.4.3 Apply simple transformations

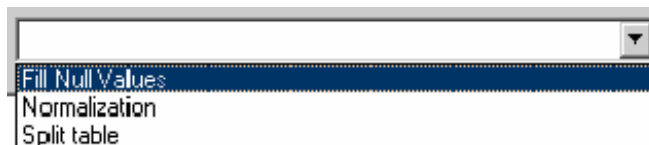
Prerequisites:

- one version is selected
- at least one simple module is loaded

Simple transformations may be applied repeatedly. Application is switched to Edit mode after the first usage of “Apply ...” button for selected version. It is possible to apply simple transformations in this mode. Changes on data are shown in DataGrid. If Knocker application is in the Edit mode then description “Data from version called:” contains both the version name and “(EDITED by Simple transformation)” note.

1. Select and apply simple transformations

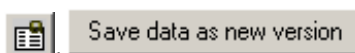
- a) Select exactly one transformation from the list of all transformations.



- b) Click on „Apply ...“button.

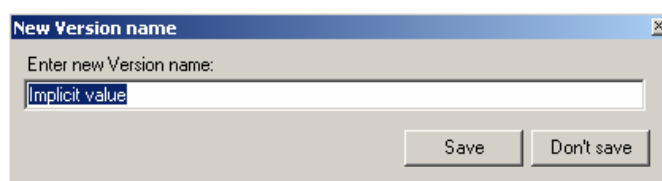
Steps a) and b) can be repeated more times.

2. Save transformed data as a new version



To save your transformed data click on „Save data as new version“button.

Enter the name for this version in the dialog.



Confirm the name by click on „Save“button.

New version will be hanged below the source version.

Note:

- Some transformations can invoke “Save data as new version” immediately after finishing their work. These transformations usually create more than one data version.

4.4.4 Apply simple transformations on other version

Prerequisites:

- version is created by simple transformations application
- another version exists

1. create a version using simple transformations
2. click right mouse button on this version and then select “Copy transformation” from context menu or open item in menu “Version -> Copy transformation”.
3. select version to which the copied transformation should be applied and then select “Apply transformation” from context menu or select “Version -> Apply transformation” in menu

“Apply transformation” can not be applied to the version with different columns. This version must have the same columns (name, simple type) as the starting version from step 1.

4.4.5 Run data mining method

Prerequisites:

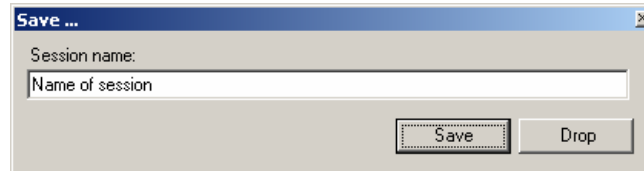
- at least one version of data
- at least one module of type “normal”

1. Choose one of the existing versions
2. From menu: Modules -> menu item with relevant Module name

4.4.6 Save session

Session is saved automatically when user creates a new one, opens an existing or selects menu item Main-> Close session.

User enters session name before saving:



If „Drop“ button is pressed, session and all included versions will be deleted from database.

4.4.7 Create new session

Main Windows menu: Main-> New.

Current session is closed before the new one is created. User is asked to save the new session whenever it contains at least one version.

5 Solution of possible problems

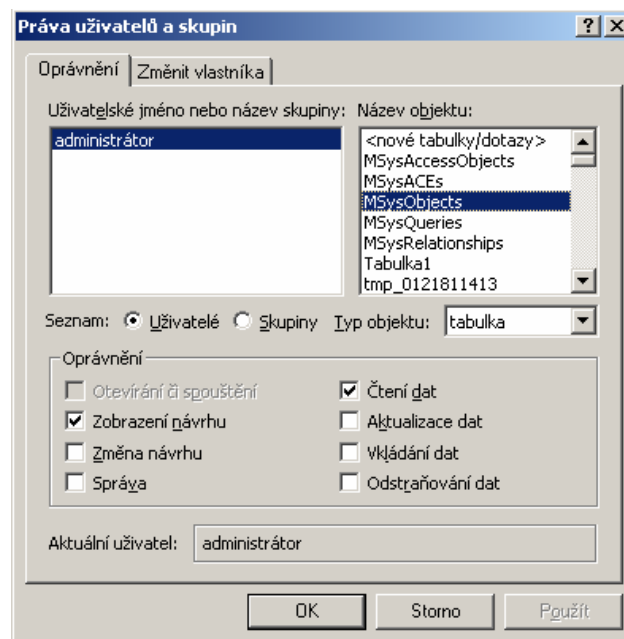
5.1 Can not connect to MS Access database (SQL Error)

Situation:

- File xxx.mdb is created
- ODBC driver lead to xxx.mdb
- Test connection is OK
- There is a SQL error in the test connection

Solution:

- Open xxx.mdb (file with database)
- MS Access menu: Tools -> Options ... -> View -> check "Hidden objects"
- MS Access menu: Tools -> Security -> Users and groups permissions -> set according this screenshot



- Try to connect again