Requirements and
Tests
Management
Repository

User guide
English Version 1.0

Contents

Document history

Date	Version	Description
2012/10/15	1.0	Document creation

Introduction

The purpose of this document is to provide detailed assistance for optimal use of the software testing tool RTMR
It is for all age groups with and without knowledge of testing methodologies.

As a reminder, RTMR stands for Requirements and Test Management Repository Repository.
This is a test tool software that allows :

- manage software requirements throughout its life cycle
- describe the scenarios and test cases to ensure the validation of these requirements
- run targeted campaigns of tests
- follow all faults and bugs meet during tests :
  - through internal bug tracker
  - through external bug tracker (bugzilla, mantis, redmine...etc)

The solution includes a version management system for projects, for requirements, for scenarios and test cases that keeps track of software changes and can easily cover all regression tests.

Terminology

Requirement : defines a characteristic of a software product imposed by a mandatory, indispensable need.

Use Case : Represents the interaction between the user and the system through various scenarios to achieve a business objective.

Test case : A set of actions to verify and validate the correct behavior, implying the expected behavior, of a function of software.

Scenario Test : set of test cases to ensure coverage of one or more requirements

regression testing : To ensure that test cases are consistent with the requirements after a software update (new version changes), This can cause degradation in using the software, or anomalies.

Recipe Booklet : document that contains comprehensive and detailed description of all test scenarios for software.

Dataset : All information (data entered, settings ... etc.) of software at a T time.

Test campaign : Defines a set of test scenarios previously selected by various criteria (criticality, priority, target ... etc.).

Each test campaign can be executed multiple times, each time defining an iteration, it can run the predefined scenarios from different data sets.

Recipe user : Phase to ensure that the software meets the business needs of users.

Bug / Problem : software malfunction identified by inconsistent conduct with the expected behavior.

Bugtracker : Software / system referencing abnormalities (Bugzilla, Mantis, Jira for the most known)

Overview

To ensure software quality through extensive coverage requirements, the R.T.M.R tool is based on all kind of actions related to software testing :

- requirements gathering
- management of tests strategy
- constitution of test cases (functional testing, performance testing, load testing etc ....)
- execution of tests

It comprises also a requirements repository and tests repository for multiple projects and multiple versions.

General operating principle of the tool is:

R.T.M.R project correspond to a software to test.

Each version of the software must meet a set of functions (functional requirements, technical requirements... etc.).
Each version of the software is linked to a version of a project into RTMR. This one is the functional memory of the application to the designated version.

Initially, the administrator of R.T.M.R create a project and its first version. It must assign permissions for different users authorized to act on this new project.

The tree of requirements is fed during gathering needs..

Scenarios and test cases must be fed in this first version and the test campaigns must be created according to the level of priority of requirements, according to the requirement types (functional, performance, load ... etc.), according to the the level of risk ... etc..

When the new version of the software is delivered, the campaigns of tests are performed to check the compliance of the software that came with the submitted requirements.

Every action of a test case is associated with a description and an expected result. If during the execution of an action, the result is not consistent, one or more abnormalities may be entered through the module for this purpose.
The execution of the campaign can generate a comprehensive report containing in one hand a summary of results from the execution of tests and on the other hand it traces step by step actions with their rate of success and failure.

For each new delivery, an administrator creates a new version of the project. In its initial state, this new version is an exact copy of the previous version, thus including the same requirements and the same test cases.
In the new version, adding, modifying or deleting requirements relating to changes (adding new features, bug fixes, performance improvements ... etc.), can target testing efforts based on related scenarios and therefore provides coverage of regression testing.

Accessing the repository

Definitions

Connection

System access requires entering a user name and password and the name or address of the gateway server : this defines the access connection.

The management of this login (user / password) can be integrated into RTMR system (the default), or may be provided by an enterprise directory supporting LDAP.

Roles

Three major role users are available :

- administrative rôle : the administrator ensures setting permissions for reading and writing for each user and for each project.
  According profiles, user definable rights are:
  - Project Access :
    - Not authorized
    - Authorized:
      - Accesses Repository requirements:
        - Unauthorized
        - Read Only
        - Read / Write
      - access repository of test cases:
        - Unauthorized
        - Read Only
        - Read / Write
      - access test campaigns:
        - Unauthorized
        - Read Only
        - Read / Write
      - Access executions test campaigns:
        - Unauthorized
        - Read Only
        - Read / Write
- writer role : he is authorized to add and modify the repository datas
- reader role : he is only allowed to read data from the repository and he can not change them

To manage permissions refer to the section Users management.

Sessions

Access to the system from connection to disconnection defines a user session.

Each session allows simultaneous access to multiple projects.

The lifetime of a session is infinite.

The system has a way of automatic reconnection in case of interruption of the connection (network outage, stopping the service server, database HS ... etc.).

Thus, the recording of data changes after loss of connection may be postponed until the connection to the remote system is restored.

Concurrency

The system allows simultaneous access to the same project and its content (reference requirements, reference scenarios, test campaigns, executions campaigns ... etc.).

Data access is managed by a locking user system. Datas can not be changed if they are already being edited in another user session.

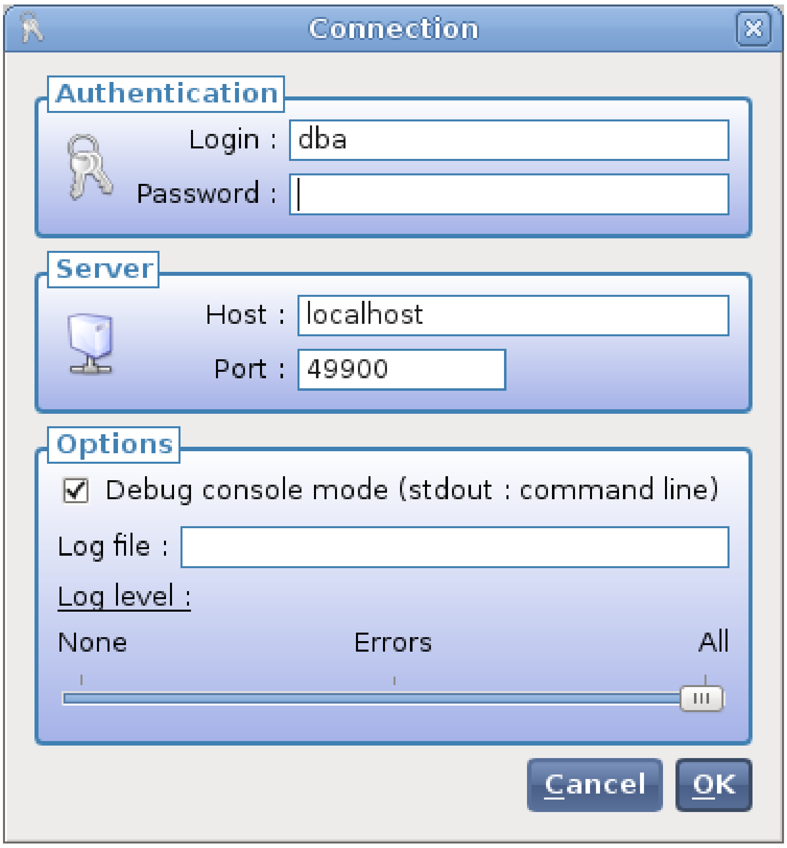
Connection

Connection screen

Launch the application according to the runtime environment :

- on Windows, double-click the rtmr.exe icon
- on MacOS, double-click the rtmr.app icon
- on Linux double-click the rtmr icon.

The login screen appears automatically:



The following informations are required to access the system :

- **Login :** user name
- **Password** : user password
- **Host :** address (IP) or name server (DNS)
- **Port :** listening port number (default 49900)

Enter those informations then, click on the [**OK**] button.

<u>Options :</u>

The **debug console mode** displays traces of execution if the application is running in console mode.

The **log file mode** records the execution traces in a file (eg log.txt)

The log level defines what kind of traces will be displayed (in console mode) or written (in file mode).

Logging modes are not exclusive.

Fisrt connection

To access for the first time to the repository, enter following informations into the logon screen :

- login : **dba**
- password : **manager01**
- server : address (IP) or server name (DNS) hosting the netserver service
- port : listening port number (default 49900) of the netserver service

These initial informations are associated with the user having a role of system administration.

It is advisable to change the administrator password from the first connection from the menu [Tools] [Change Password].

Once logged in as dba (administrator), the Administration menu allows you to create users and assign access rights to users for each project.

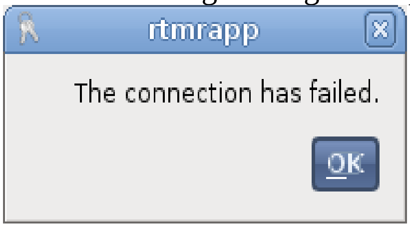
For more informations on access management refer to the section Users management.

Creating projects is only accessible to the administrator.

Connection errors

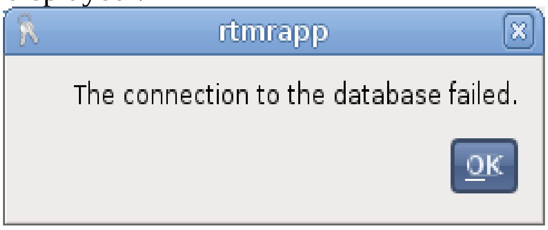
The following errors are possible when attempting to connect to the server :

- If the server does not respond the following message is displayed :

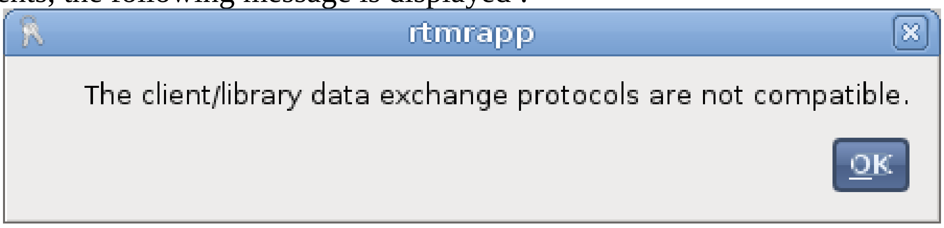


Check the address and port number of the server before and if this informations are correct contact the administrator to ensure that the server works.

- If the database does not respond or if the name and/or password entered are incorrect the following message is displayed :



- If the client (rtmrapp) protocol and the library (librtmr) protocol used for data exchange are differents, the following message is displayed :



- If the server protocol and the client protocol used for data exchange are differents, the following message is displayed :

To achieve connection, an update of the client or the server is needed.

Check the client version installed from the menu [**?**][**About**].

Check the server version installed with the server administrator (command line : netserver -v).

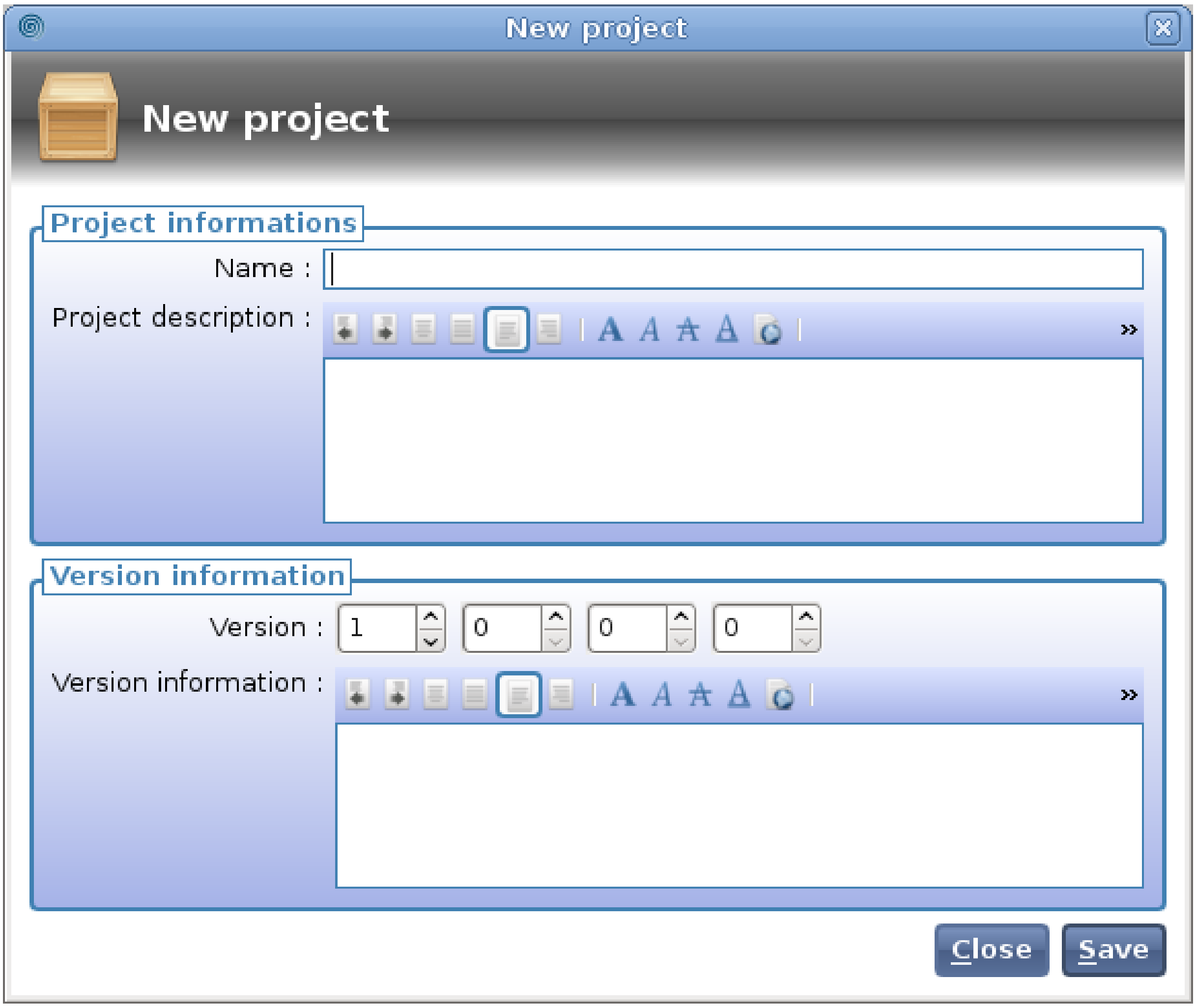
Projects management

Add a new project

Prerequisites : Be connected as an administrator (see Fisrt connection or Users management)

To create a new project, select the menu [**File**][**New project...**].

The following screen is shown :



Enter the following informations :

- **Name** : enter the name of the new project, it could be the name of the software to test (e.g : Rent Software)
- **Project description** : enter the description of the project, it could a brief description of the software to test (e.g : This software is used to manage car renting).
- **Version** : enter the version number of the software to test (eg : 3.4.0.0). The rule is :
  - 1st number : version number link to major changes
  - 2nd number : version number link to minor changes

– 3rd number : version number link to corrections of bugs

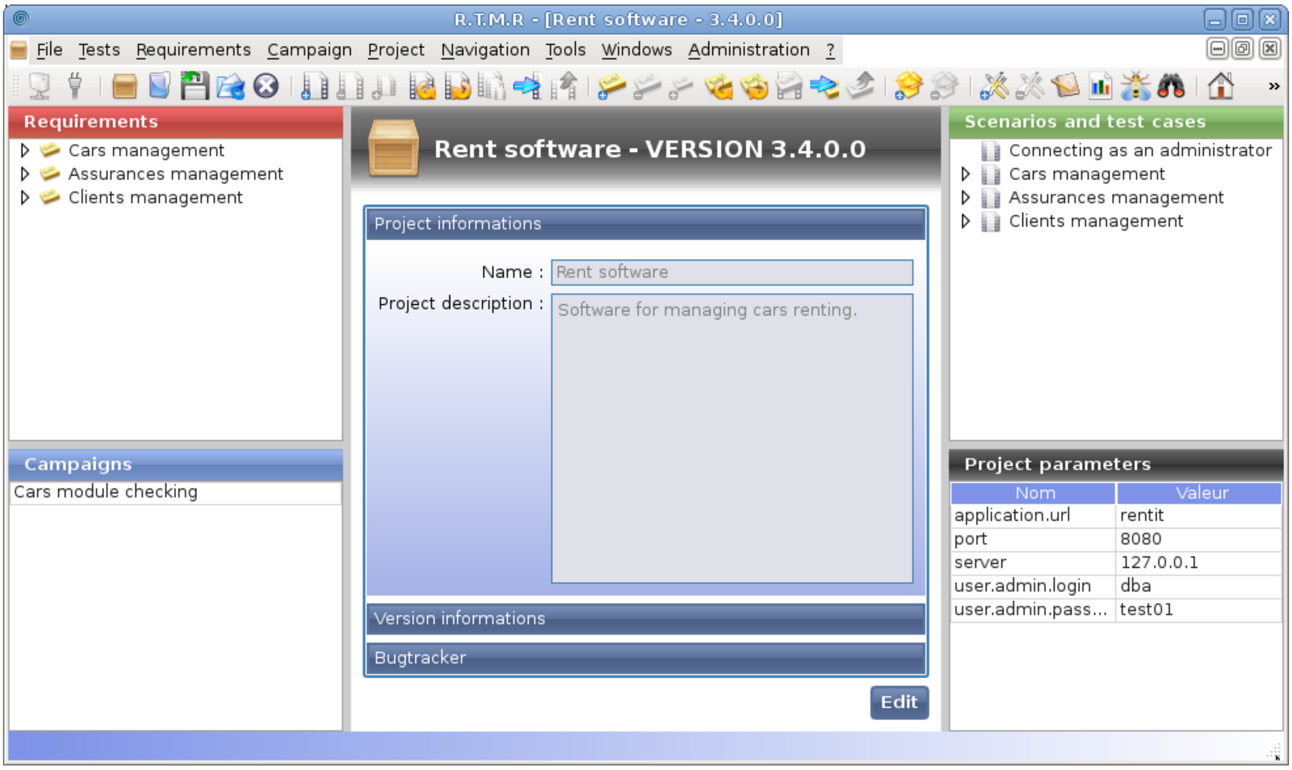
– 4th number : internal version number

– **Version informations** : enter the description of the features of this version (e.g : This version 3.4 has a client module, a garanty module,...etc).

Creating a new projects implies automaticly the creation of a first version. This fisrt version is not necessary the first version of the software to test.

Click on the [**Save**] button to create the project and its first version.

The main windows of the project is displayed :



On the upper left corner there is the requirement tree (see Requirements management).

On the down left corner there is the list of test campaigns (see Campaigns management).

On the upper right corner there is the tests tree (see Management of scenarios and test cases).

On the down right corner there is the list of project parameters (see Manage project parameters).

On the center there is the screen of the project properties.

Import a project

Prerequisites : Be connected as an administrator (see Permissions management).

Select the menu [**File**][**Import a project**], then select a xml file compatible with the format of R.T.M.R files.

The imported project name will be suffixed with the date and hour of the import.

The name of the project would be change later : see Modify project properties.

Export a project

Prerequisites : Have a project opened and have access rights for reading on to the requirements module and on to the tests module (see Permissions management).

Select the menu [**File**][**Export a project**], then save the file on to the selected destination.

The exported file is a xml file that contains :

- project informations
- project parameters
- project versions informations :
  - requirements
  - scenarios and test cases
  - tests campaigns

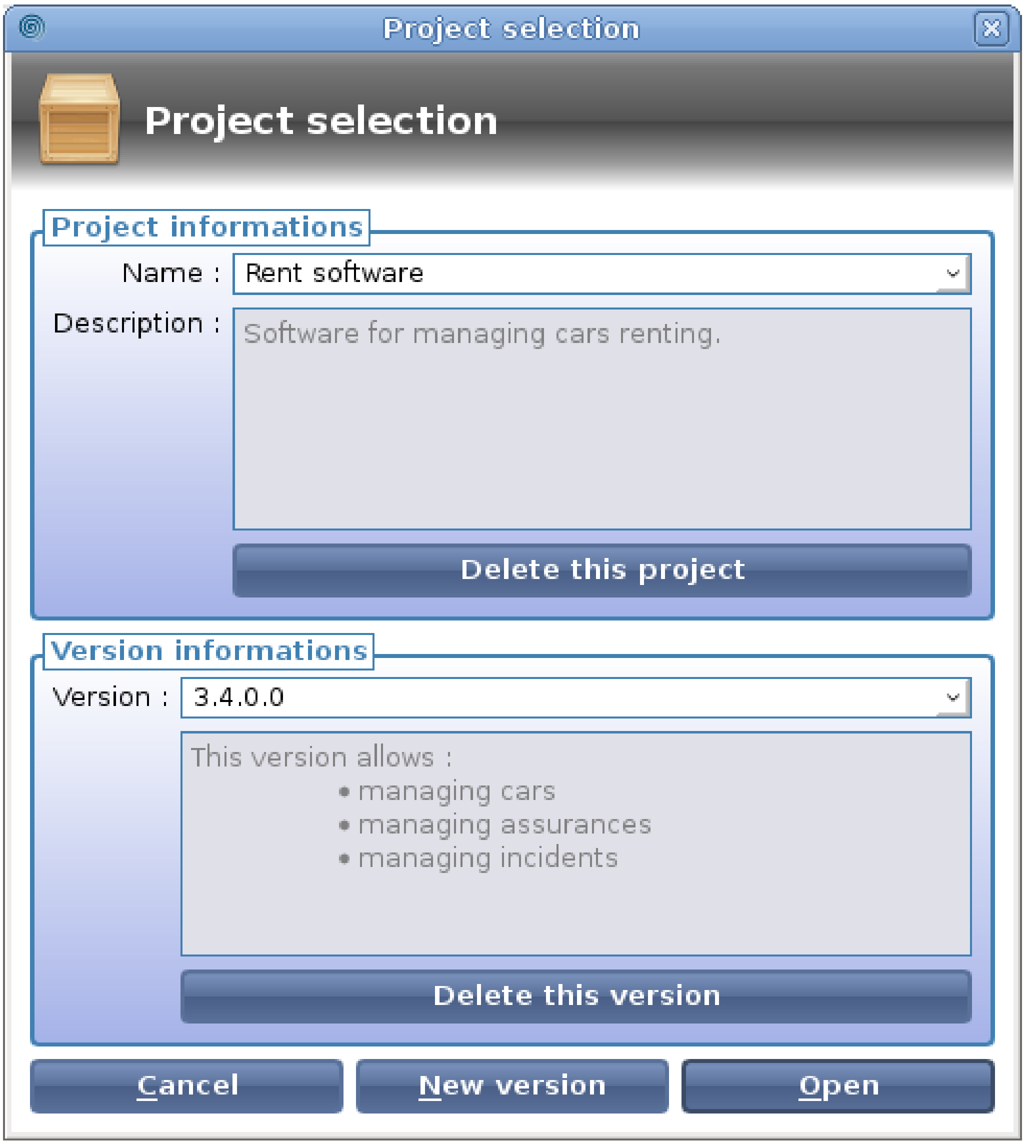
Campaign runs are not exported.

Open a project

Prerequisites : Be connected with project access rights (see Permissions management).

Select [**File**][**Open a project**].

The following screen is displayed:



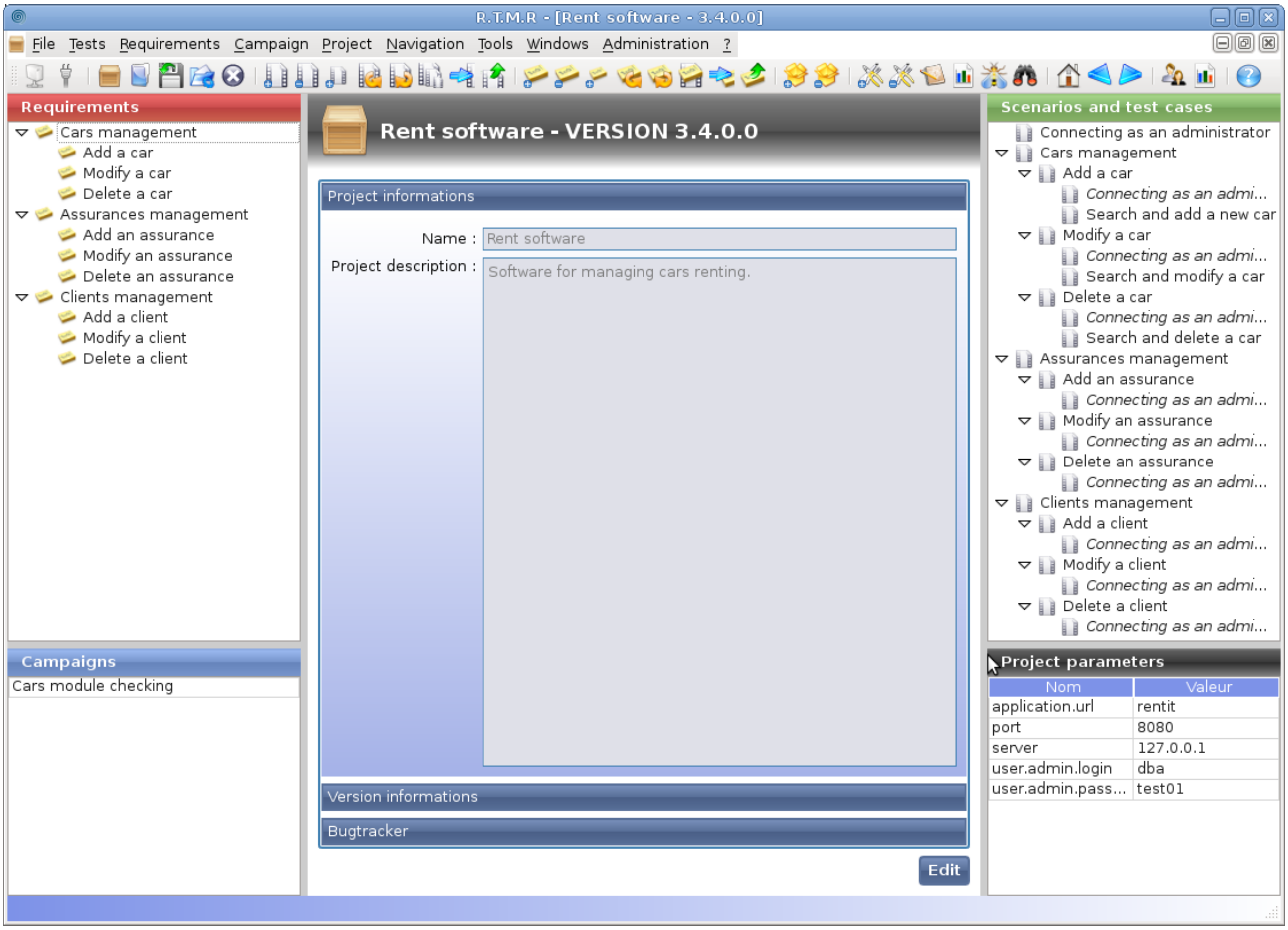
The listbox **Name** inditifies all acessibles projects.

The listbox **Version** inditifies all acessibles versions for selected project.

To open a project, select the project and its version then click on [**Open**] button.

P.S : Opening a project is inseparable with opening a project version. That is to say, a project is always opened in a certain version.

The main window of the project is displayed :



On the top left there is the requirements tree (see Requirements management).

On the bottom left there is the tests campaigns list (see Campaigns management).

On the top right there is the tests tree (see Management of scenarios and test cases).

On the bottom left there is the project parameters list (see Manage project parameters).

On center there is the project properties panel.

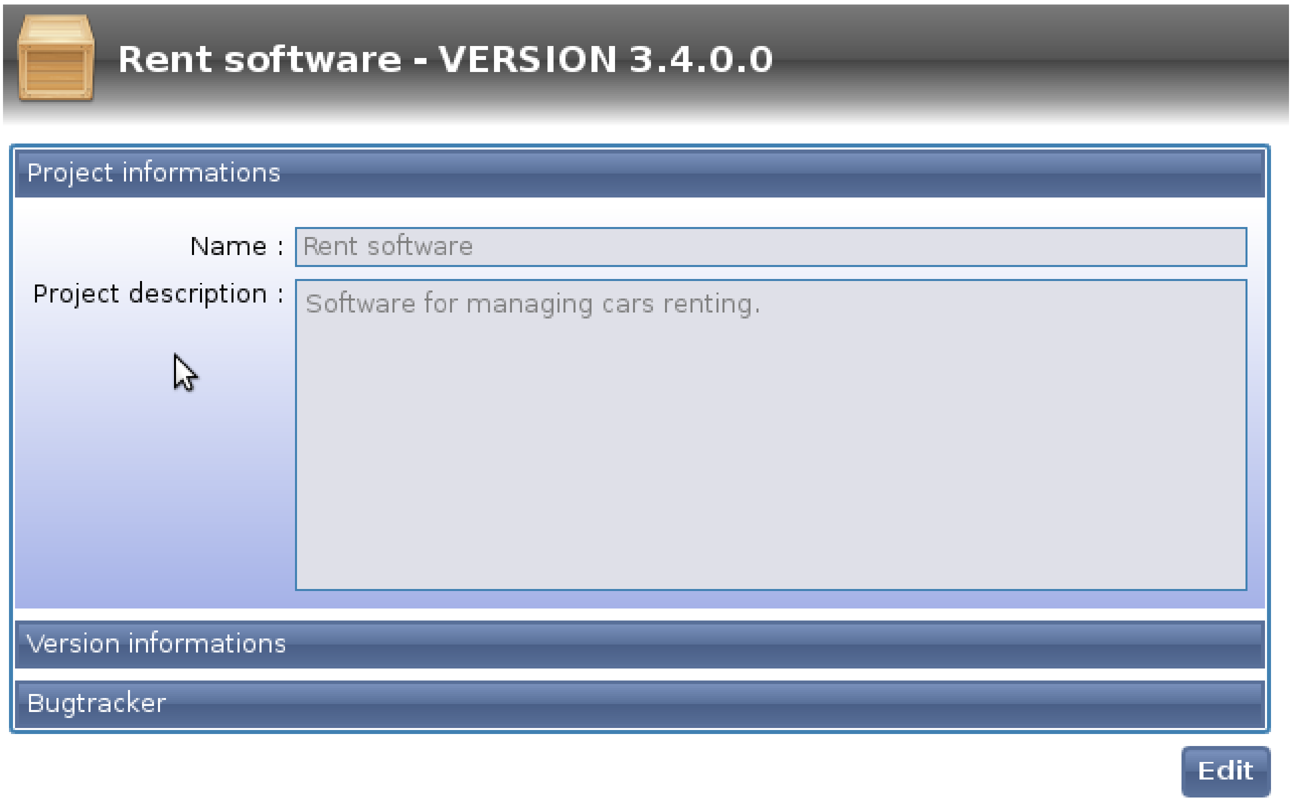
Modify project properties

Prerequisites : Have a project opened and have a writer role (see Permissions management).

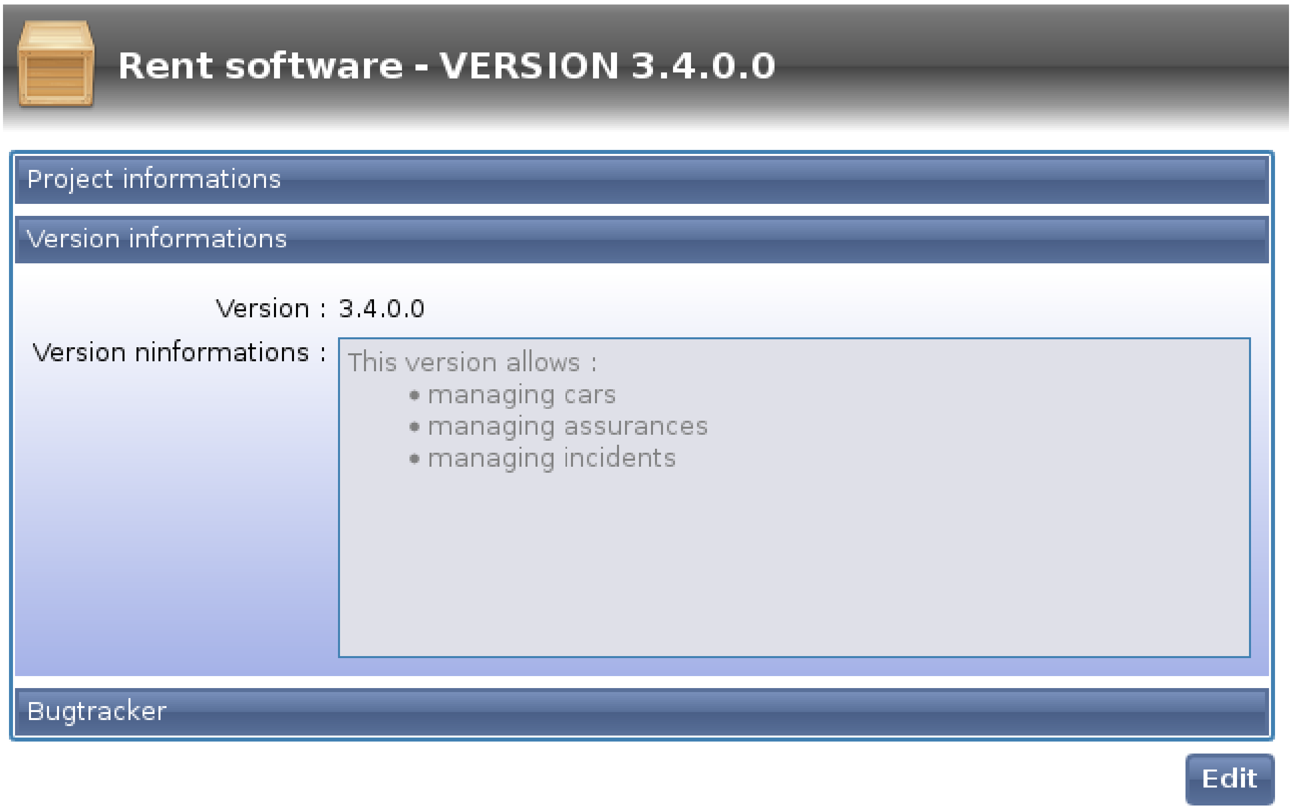
Open a project (see Open a project).

Select the menu [**Project**][**Project properties**].

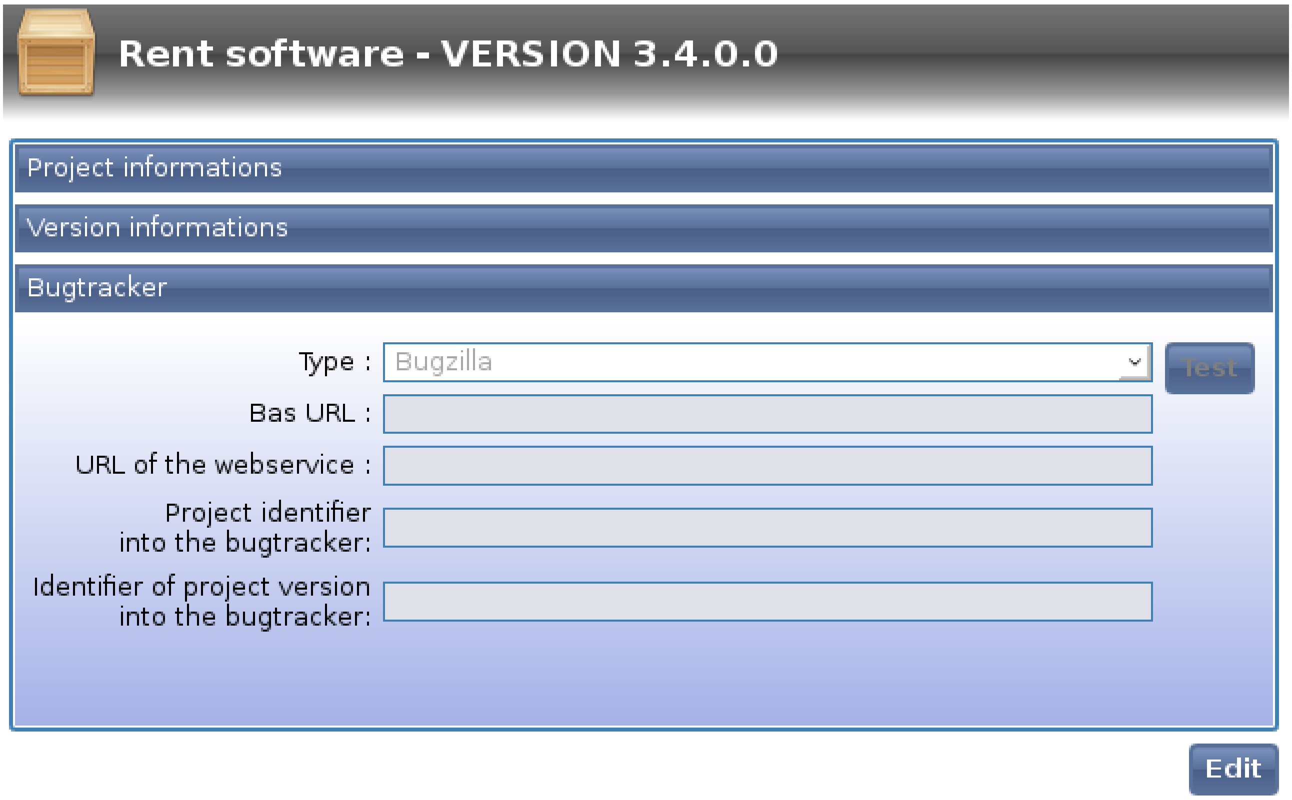
On the center of the main windows propect properties screen is displayed :



Click on the version information box to see :



Click on the Bugtracker box to see :



Press the [**Edit**] button to modify :

- project name
- project description
- project version informations :

- version description
- datas for connecting to a external bugtracker if it exists :
  - bugtracker type (mandatory)
  - Base URL (optional) : this is the address you use to connect to the bugtracker from your browser. This address is used to display web links pointing directly to the description of an anomaly into the bugtracker
  - Webservice URL (optional) : depend bugtracker type (see below)
  - Project identifier (optional) : this is the identifier of the project into the bugtracker
  - Version identifier (optional) : this is the identifier of the project version into the bugtracker

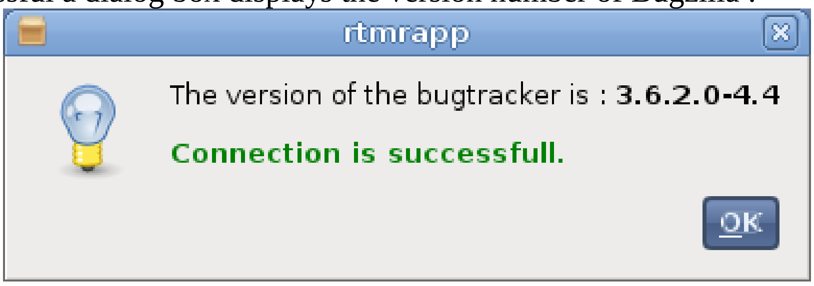
To properly use the external bugtracker, define access settings from the menu [**Tools**][**Options**], then select the tab [**Bugtracker**].

Configuration of Bugzilla

Set the following properties :

- bugtracker type : Bugzilla
- Base URL (mandatory) : this is the address you use to connect to Bugzilla from your browser (eg : http://localhost/bugzilla)
- Webservice URL (optional) : Put the base URL followed by /xmlrpc.cgi (eg : http://localhost/bugzilla/xmlrpc.cgi
- Project identifier (optional) : this is the identifier (numeric) of the Bugzilla product (eg : 12). This identifier is optional if the RTMR project name is identical to the Bugzilla product name, the comparison between those names is case sensitive.
- Version identifier (optional) : this is the identifier (alphanumeric) of the Bugzilla project version. This identifier is optional. however when entering a new bug, a check is made between Bugzilla project version and RTMR project version, the comparison between those versions is case sensitive.

When informations are entered click on [**Test**].
If the test is successful a dialog box displays the version number of Bugzilla :



If the test fails, several reasons are possible :

1. If the Bugzilla server is outside your network and you usually go through a proxy : you may configure your proxy through the menu [**Tools**][**Options**] then tab [**Proxy**].

2. To operate, the Bugzilla webservice needs the following php libraries :
   - SOAP Lite
   - JSON-RPC
   - Test-Taint

Check the actual installation of those libraries using the perl script *checksetup.pl* included with Bugzilla on the server hosting the bugtracker (eg: under linux debian /usr/share/bugzilla3/lib/checksetup_nondebian.pl). If the modules are not installed, install them, either from the perl script *install-module.pl* provided by Bugzilla, or from a system package manager (under linux : yum for Redhat, Yast for OpenSuse, apt-get for Debian...etc).

Configuration of Mantis

Set the following properties :

- bugtracker type : Mantis
- Base URL (mandatory) : this is the address you use to connect to Mantis from your browser (eg : http://localhost/mantis)
- Webservice URL (optional) : Put the base URL followed by /api/soap/mantisconnect.php (eg : http://localhost/mantis/api/soap/mantisconnect.php
- Project identifier (optional) : this is the identifier (numeric) of the Mantis project (eg : 12). This identifier is optional if the RTMR project name is identical to the Mantis project name, the comparison between those names is case sensitive.
- Version identifier (optional) : this is the identifier (alphanumeric) of the Mantis project version. This identifier is optional. however when entering a new bug, a check is made between Mantis project version and RTMR project version, the comparison between those versions is case sensitive.

When informations are entered click on [**Test**].
If the test is successful a dialog box displays the version number of Mantis:

If the test fails, several reasons are possible :

1. If the Mantis server is outside your network and you usually go through a proxy : you may configure your proxy through the menu **[Tools][Options]** then tab **[Proxy**].
2. For Mantis, a recurring problem is the use of deprecated php function in the webservice that prevents proper operation.
   Mantis 1.1.X is know to act with php 4, Mantis 1.2.X with php 5 (see http://www.mantisbt.org/requirements.php).
   Check the php version from your installation of Mantis with the following command :
   $ php -v
   Then , check the version of Mantis simply by logging on the web interface.

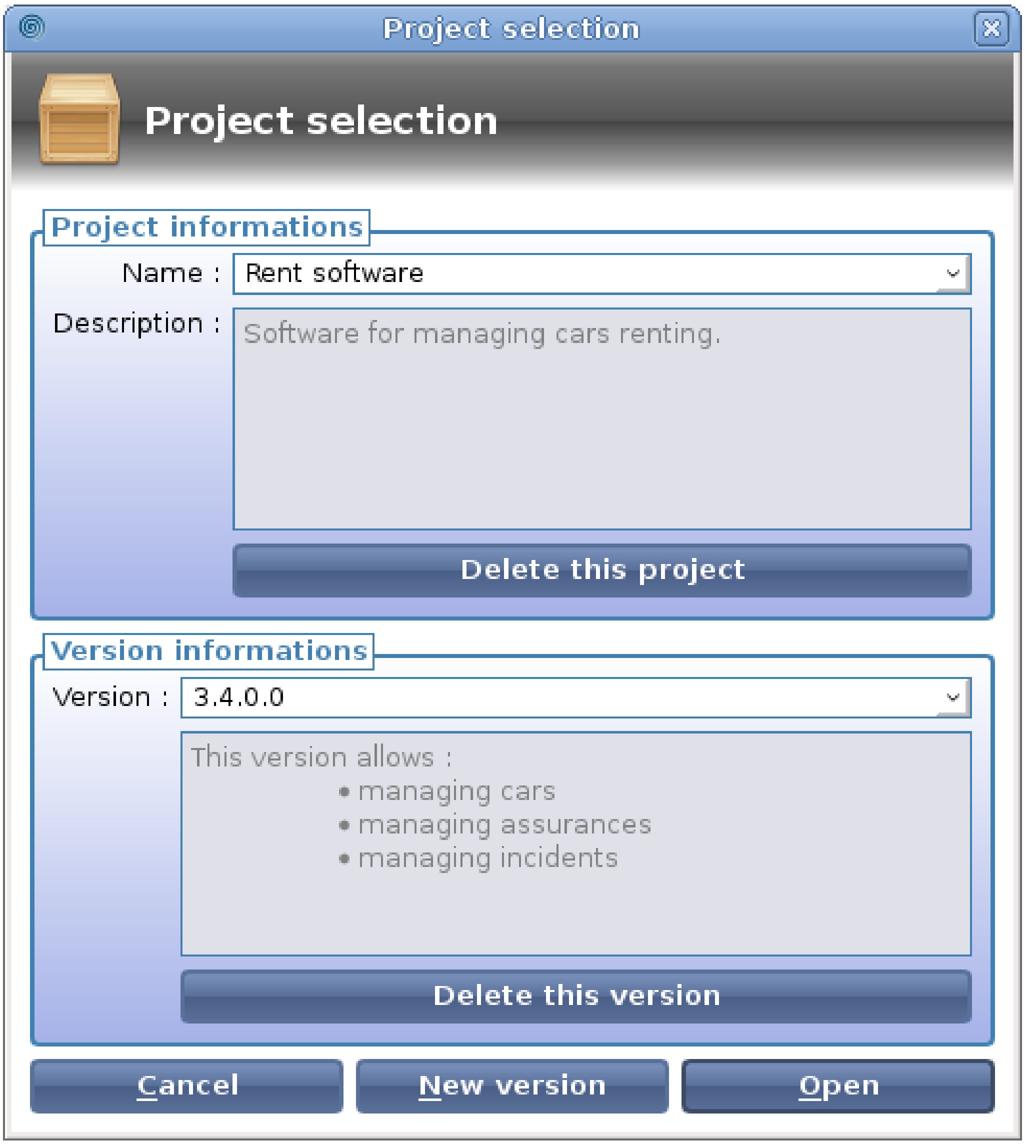
If the versions Php/Mantis are not "compatible" you need to update your system for Mantis connectivity works.
PHP is used by other services, that's why it's advised to update Mantis rather than Php. These update operations must be performed by the administrator of the server hosting Mantis.

Delete a project

Prerequisites : Have a project opened and have a administrator role (see Fisrt connection and Permissions management).

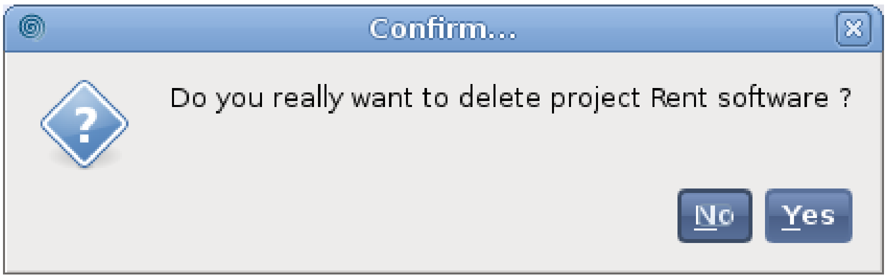
Select the menu [**File**][**Open a project**].

The following screen appears :



Name drop-down list shows the list of projects available.
Version drop-down list shows the list of versions of the selected project.

To delete a project, select the project and then click on [**Delete this project**].
A confirmation window prompts you to confirm your choice :



Click on [**Yes**] to delete the selected project else click [**No**].

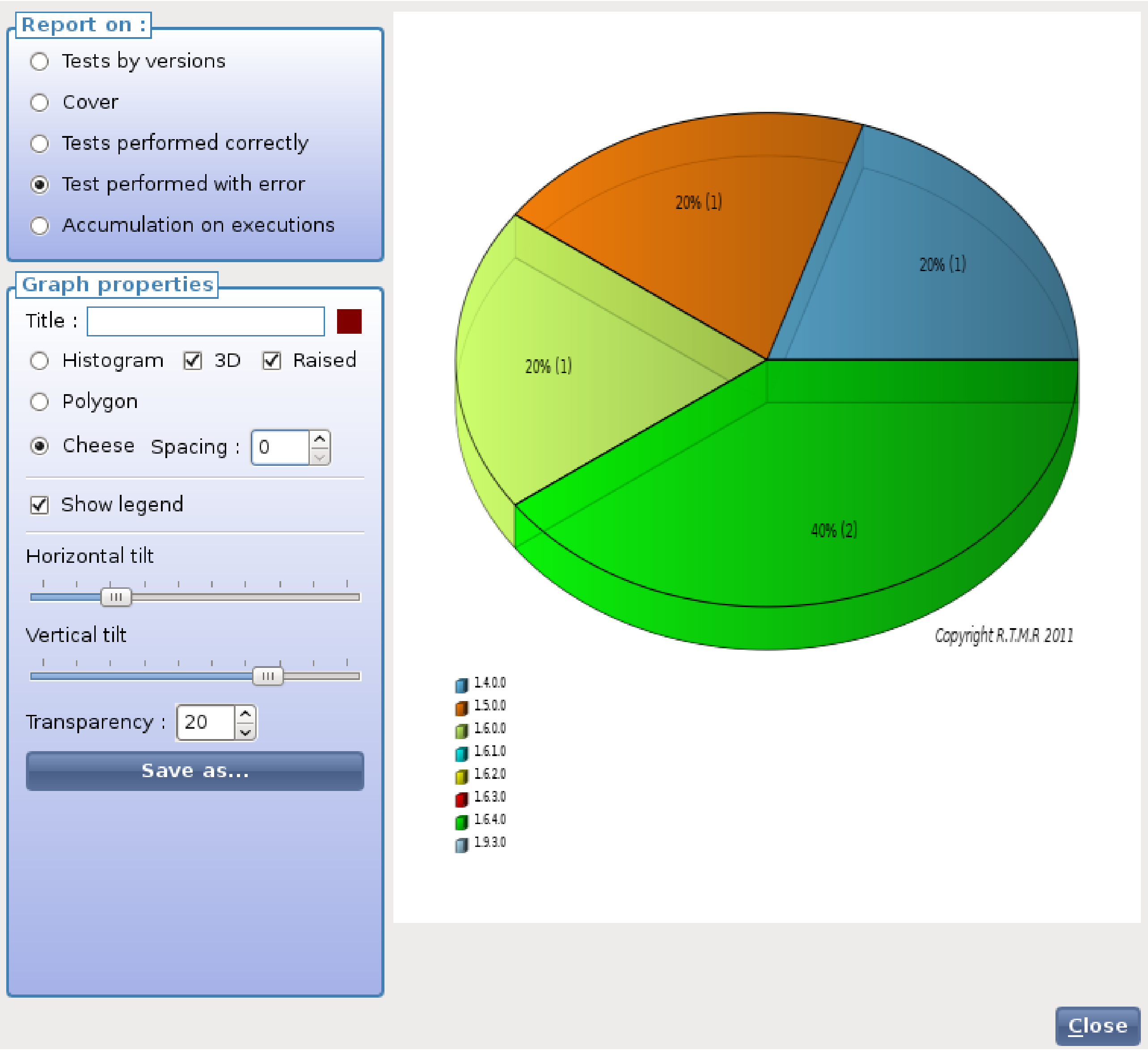
Attention : deleting a project deletes all data associated :

- all versions of the project
- all requirements
- all scenarios and test cases
- all campaigns
- all executions of campaigns
- all project parameters
- all bugs

Executions reports of projects

Prerequisites : Have opened a project (see Permissions management).

Select the menu [**Project**][**Executions reports**].



Reporting options are :

- tests par versions : displays the number of tests per project versions
- Cover : displays the number of tests covered and the number of tests not covered in the entire project for all versions
- tests performed correctly: displays the number of tests performed correctly for each version of the project
- tests performed with error : displays the number of tests performed with error for each version of the project
- Accumulation on exécutions : display a test states performed for each version of the project

Versions management

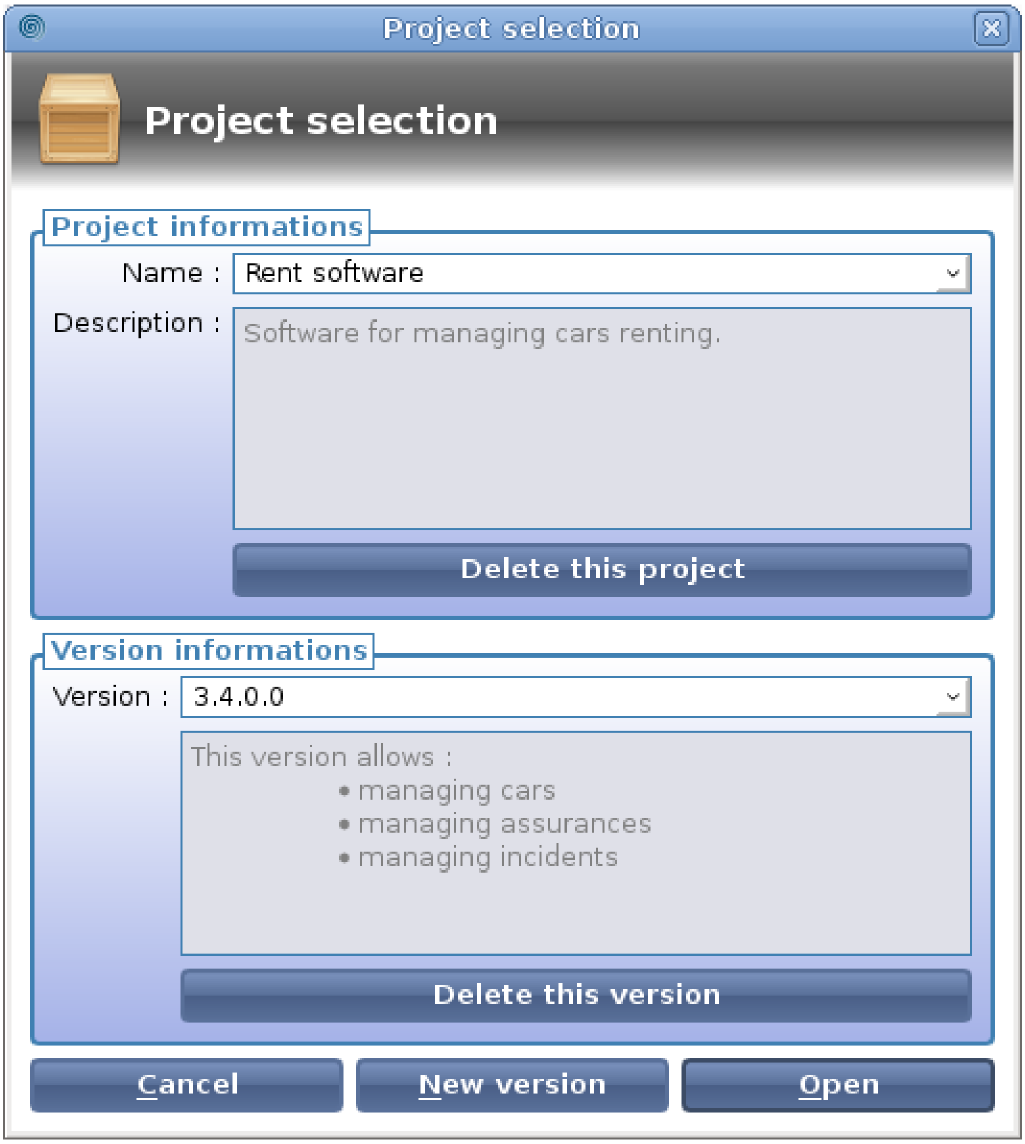
Create a project version

Any version of a project, except the first, is a copy of an earlier version.
To create the first version of a new project, refer to section Add a new project.

Prerequisites : Be connected with a user having writer role and allowed to access the project (see Permissions management).

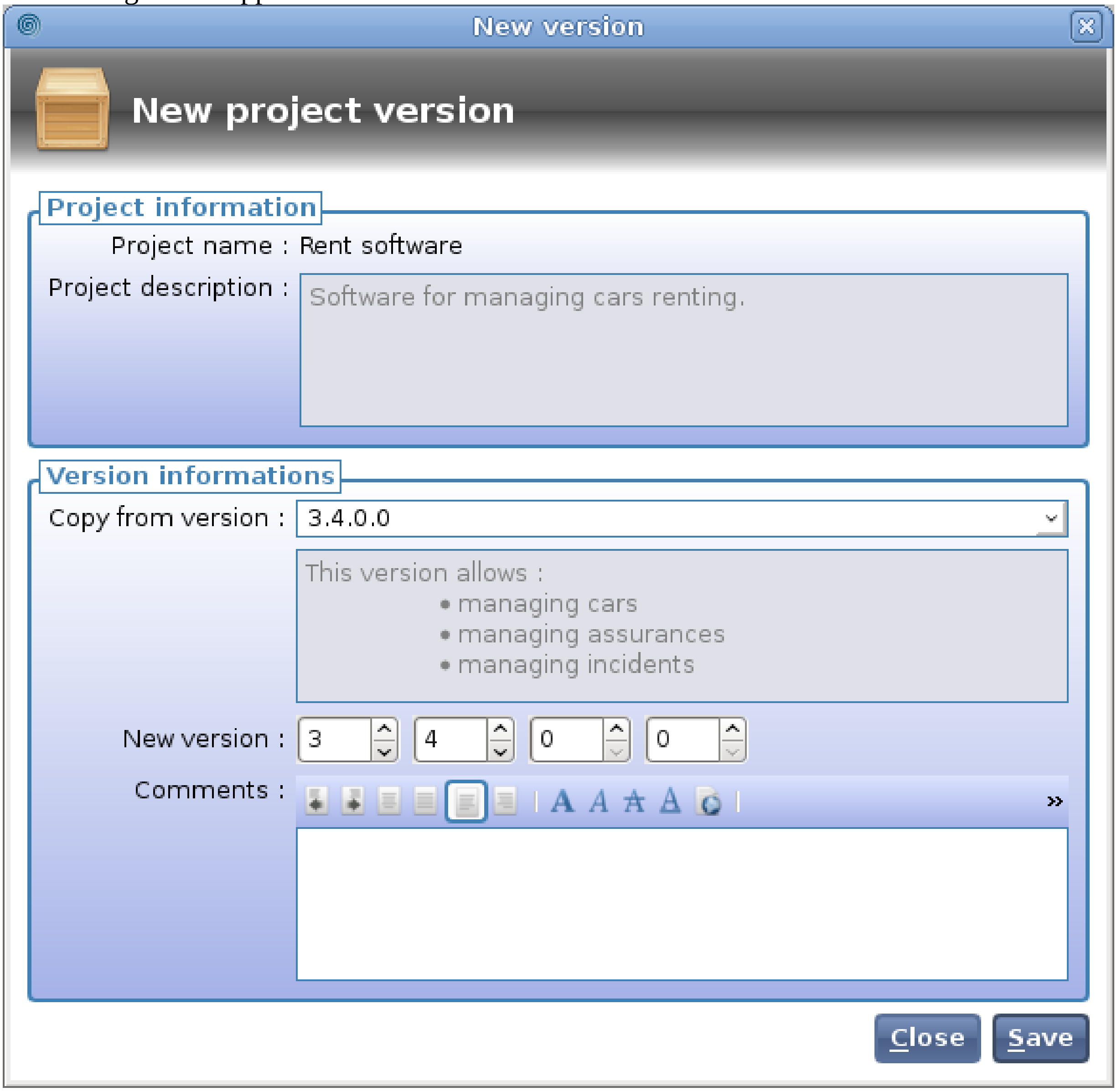
Select the [**File**][**Open a project**].

The following screen appears :



Name drop-down list shows the list of projects available.
Version drop-down list shows the list of versions of the selected project.

Select the project for which to create a new version, then click on [**New version**].
The following screen appears :



Select the source version number from the dropdown **Copy from version**.
Enter the version number of the new version in the fields **New version**.
Enter the description of the new version in the area **Comments**.

Click on [**Save**] to create a new version from a copy of the selected version.

Each copied version contains all requirements, all scenarios and all the campaigns of the original version.
The project parameters are global projects and are thus in all versions of a project
(see Manage project parameters).

Open a project version

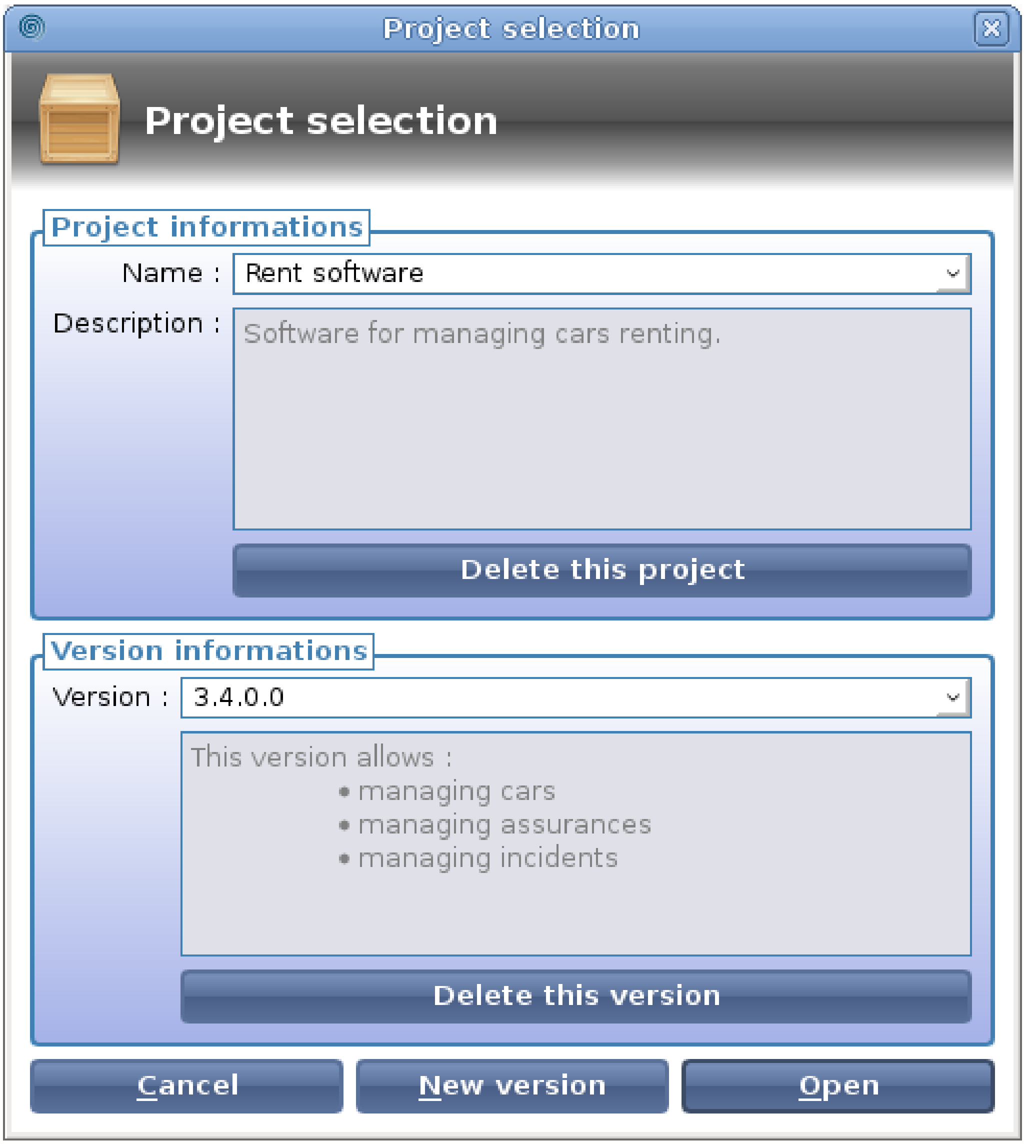
See section Open a project.

Delete a project version

Prerequisites : Be connected with a user having role administrator (see Permissions management).

Select the menu [**File**][**Open a project**].

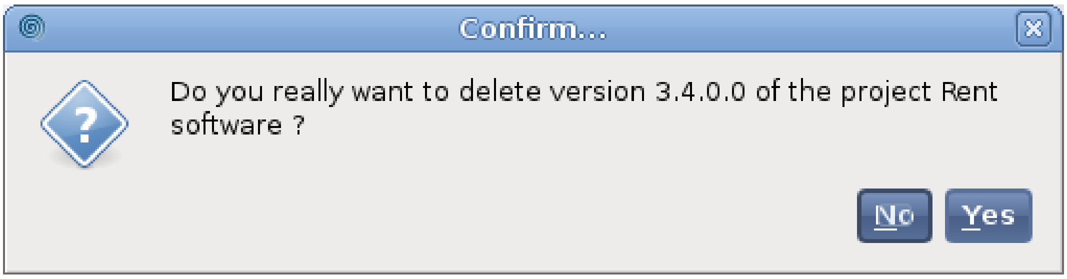
The following screen appears :



Name drop-down list shows the list of projects available.
Version drop-down list shows the list of versions of the selected project.

To delete a project version, select the project then select the version and click on [**Delete this version**].

A confirmation window prompts you to confirm your choice :



Click on [**Yes**] to delete the version else click on [**No**].

Attention : deleting a project version erases all data associated :

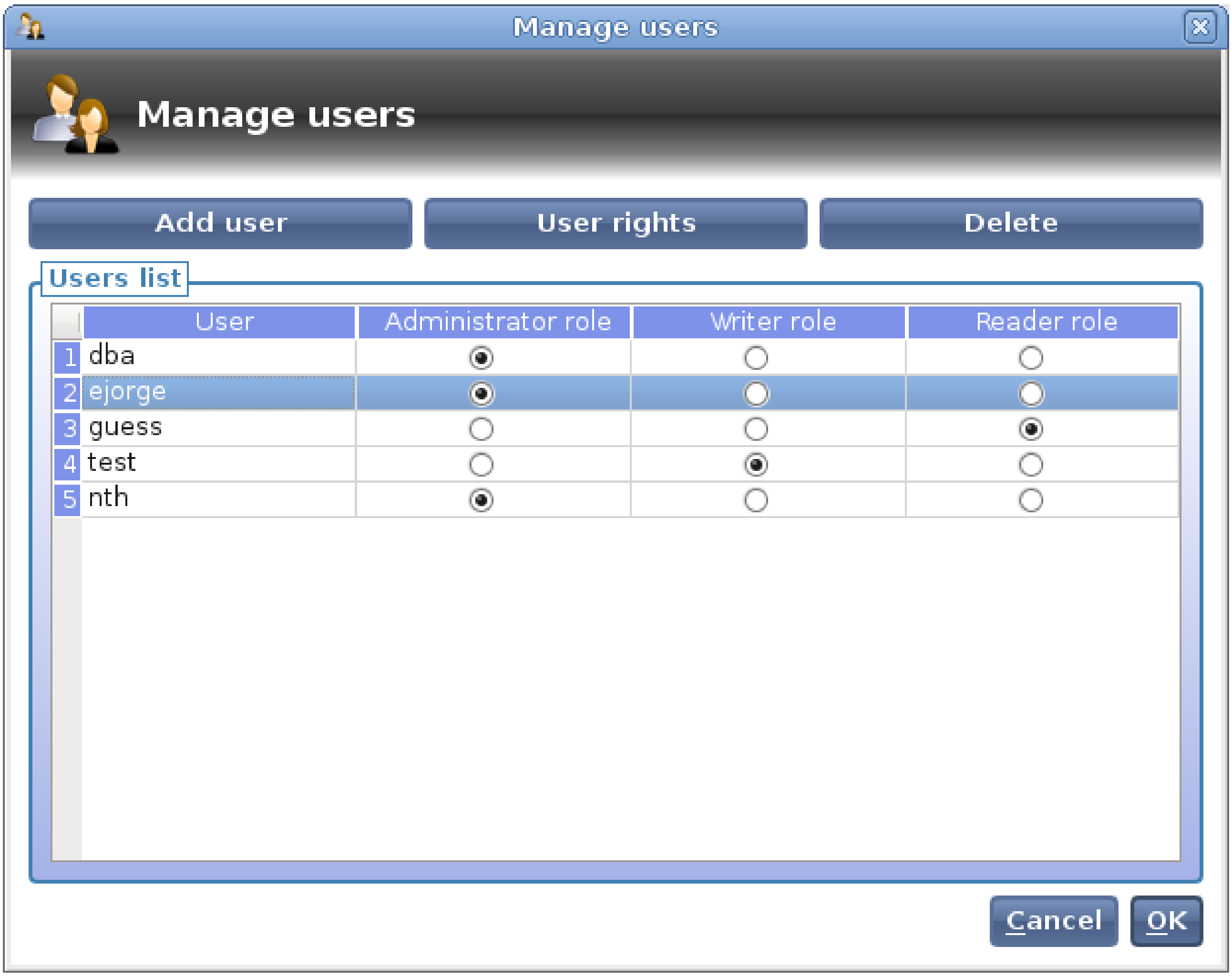
- all requirements of the version
- all scenarios of the version
- all campaigns of the version
- all executions campaigns of the version
- all bugs of the version

Users management

Prerequisites : Be connected with a user having role administrator (see Fisrt connection or Users management).

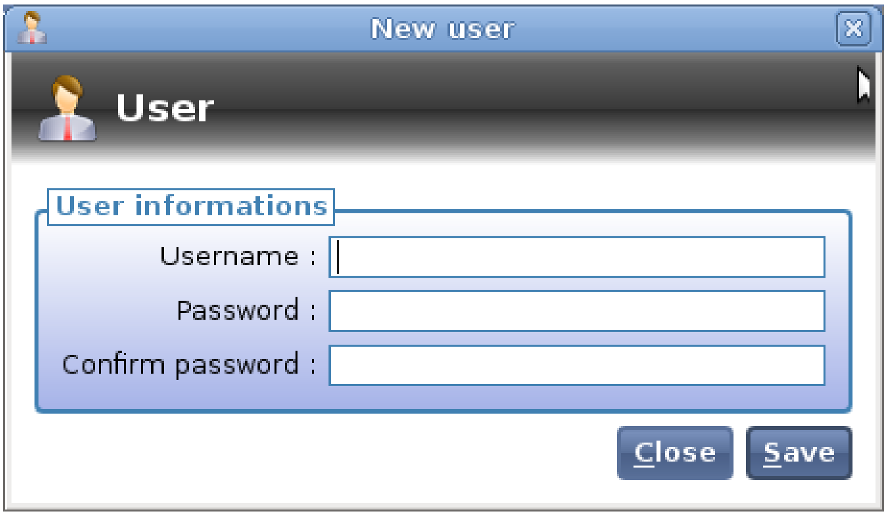
Select the menu [**Administration**][**Manage users**].

The following screen appears :



Add a user

In the user management window click on [**Add user**].
The following screen appears :



Enter mandatory informations :

- **Username**
- **Password**
- **Confirm password**

Click on [**Save**].

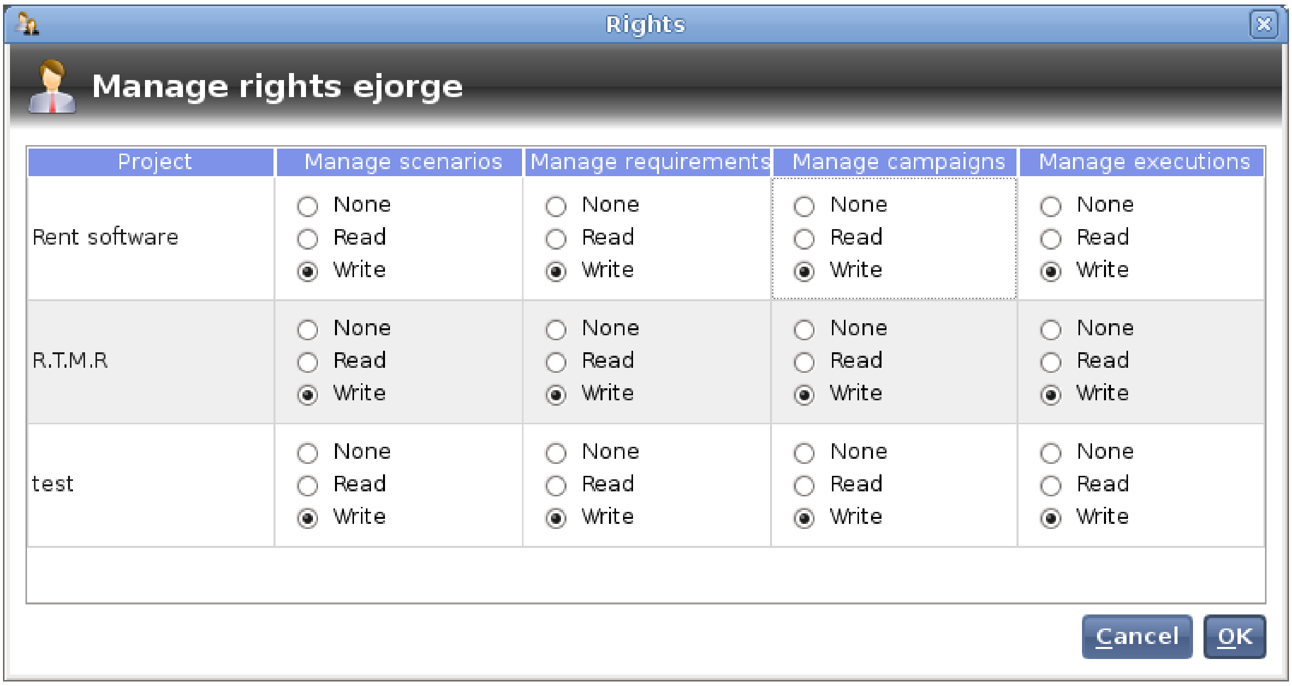
Back to **Manage users** screen, possibly clarify the role of the new user (administrator, writer or reader) and then click on [**OK**].

P.S : Each new user is reader by default.

For more informations on available roles see section Roles.

Permissions management

In the **Manage users** window select a user from the list and click on [**User rights**].
The following screen appears :



This window contains the list of projects and for each of them, the user permissions for the following modules :

- scenarios
- requirements
- campaigns
- executions of campaigns

Modify permissions selecting radio buttons [**None**], [**Read**] and [**Write**] for each project and for each of the four modules :

- **None** mode does not allow the user to view datas associated with the module
- **Read** mode allow the user to view datas associated with the module
- **Writer** mode allow the user to view, modify and delete datas associated with the module

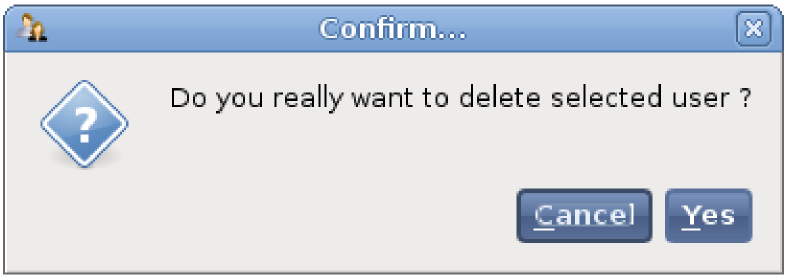
Click on [**OK**] to validate your modifications.

Back on the **Manage users** window, click the [**OK**] to validate the changes permanently.

Delete a user

In the **Manage users** window select a user from the list and click on [**Delete**].

A confirmation window prompts you to confirm your choice :



Click on [**Yes**] to delete selected user else click on [**Cancel**].

Back on the **Manage users** window, click the [**OK**] to validate the changes permanently.

Requirements management

Introduction

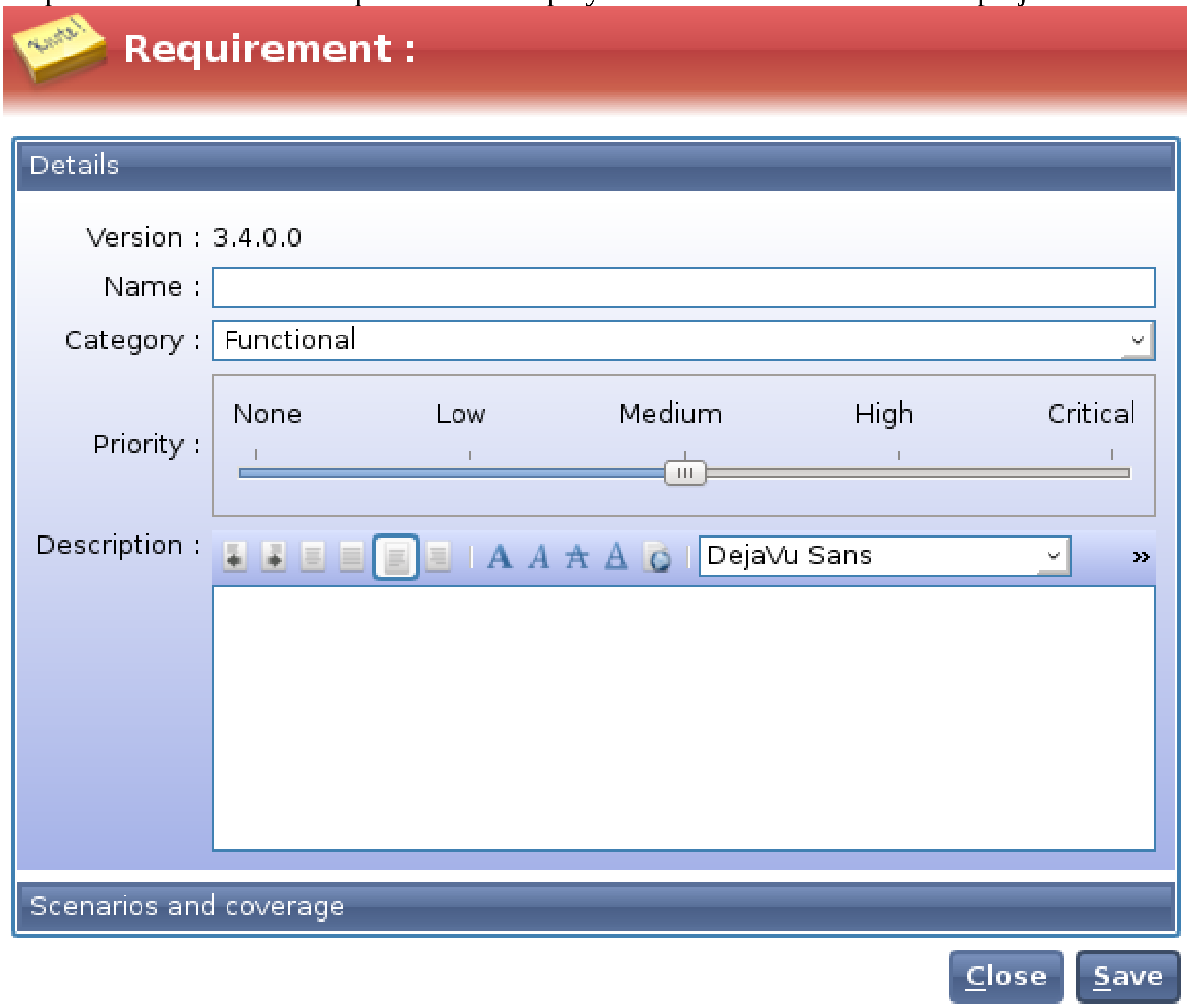
In order to structure the management of requirements, the repository of requirements is presented in the form of a hierarchical list.
The interest is to create sets of requirements group by functional module, or group by type or group by priority ... etc.
Each requirement or group of requirements can contain sub-sets of requirements called childs requirements.

Although the number of sub-level sets of requirements is infinite, it is advisable not to exceed four levels to simplify the understanding of the repository structure.

Add a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

From the main window of the opened project, select the menu [**Requirements**] [**New Requirement**] or right-click in the requirements sidebar and select [**New Requirement**].
By default, the requirement is directly created in the tree and is positioned after the selected requirement when handling.
To create a child requirement select the menu [Requirements] [**Add child requirement**] or right-click in the requirements sidebar and select [**Add child requirement**].
By default, the requirement girl is created directly in the tree and is positioned below the selected requirement when handling.
The input screen of the new requirement is displayed in the main window of the project :



The version of the requirement is handled automatically: it is the same version as the opened project version.
Enter requirement informations :

- **Name** : mandatory (eg : Manage cars)
- **Category** : this is the type of requirement or group of requirements associated with a need

to use the tested software :

- Fonctionnal (eg : manage cars of any mark)
- Interoperability (eg : import car files)
- Load : (eg : handle 50 simultaneous users)
- Performance (eg : search result of the customer database in less than 10 seconds)
- Disponibility (eg : handle connection all the year, 24/24 and 7/7)
- Security (eg : restricted access to super-user for insurers data)
- Exploitability (eg : generate daily activity reports for rental)
- Other (eg : support accessibility for disabled users)

- **Description** : that's the requirement description
- **Priority** : that's the requirement matter level
  - critical
  - high
  - medium
  - low
  - none

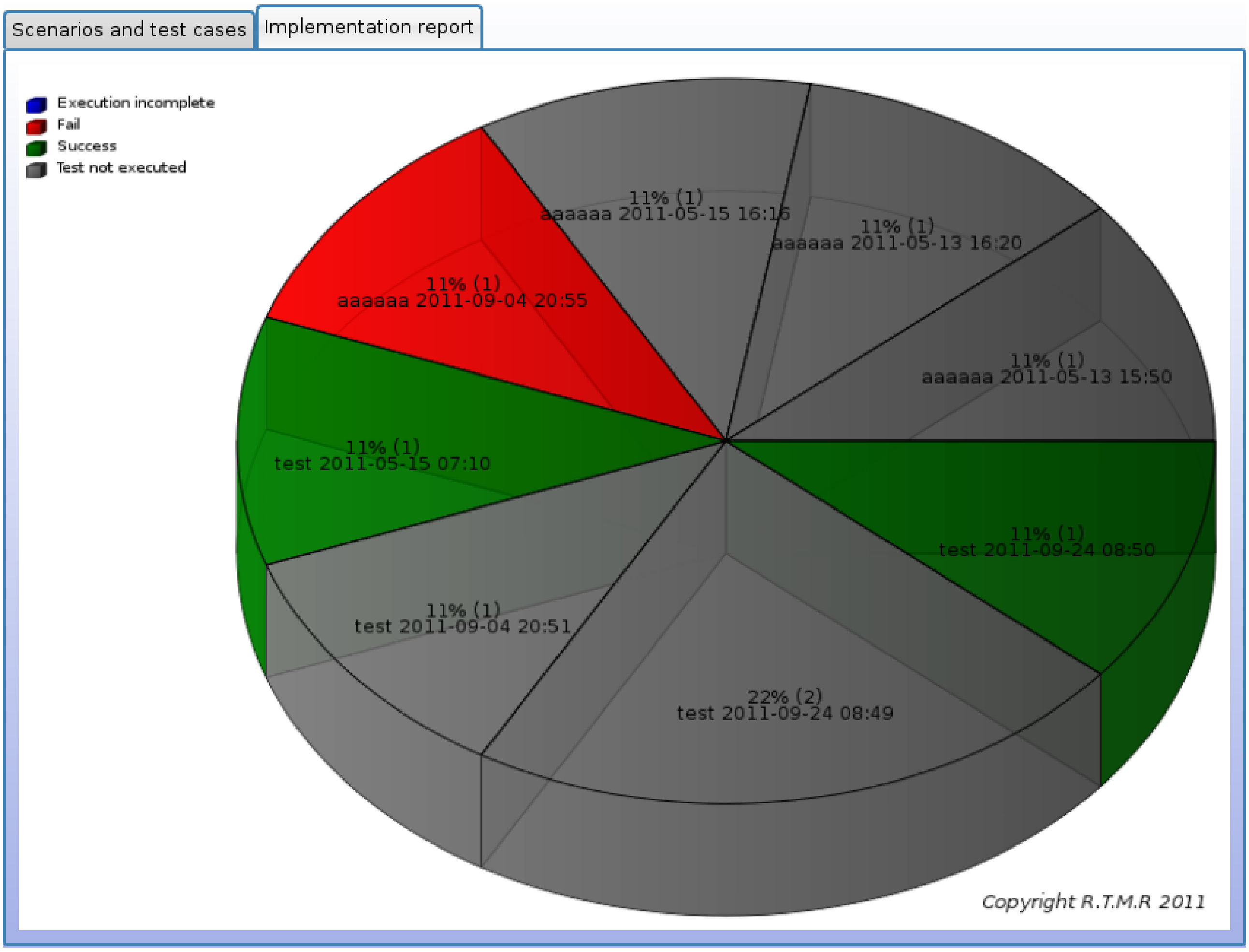
Click on [**Save**] to validate the new requirement.

Manage associated scenarios

Scenarios and test cases tab lets associate scenarios to the currently edited requirement (see Link scenarios and test cases to a requirement).

Executions reports

Execution reports tab lets display coverage statistics of the currently edited requirement :



This graphic shows the number of tests associated with the requirement which has been :

- executed correctly
- executed with error
- incompletely executed
- By passed

The repartition is done in relation to campaigns executions.

Modify a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

Left click on the requirement to modify in the left pannel.

The entry screen of the requirement is displayed (voir Add a requirement).

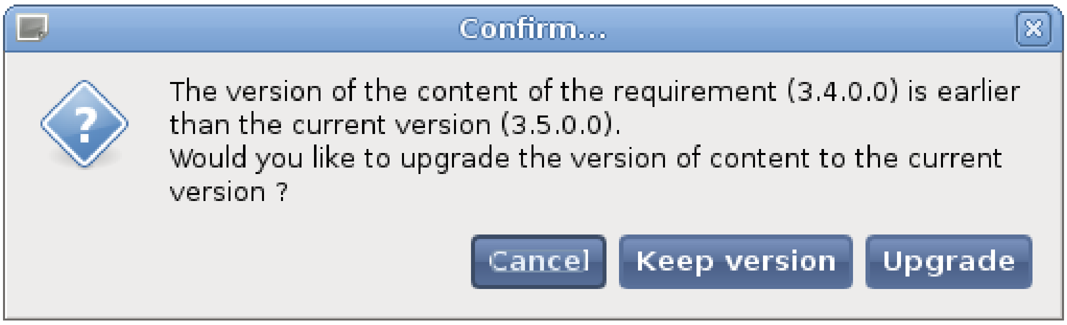
Modify requirement data then click on [**Save**].

Scenario 1 - requirement version is the same as project version :

In this case, changes are saved.

Scenario 2 - requirement version is different from project version :

In this case, the following message is displayed :

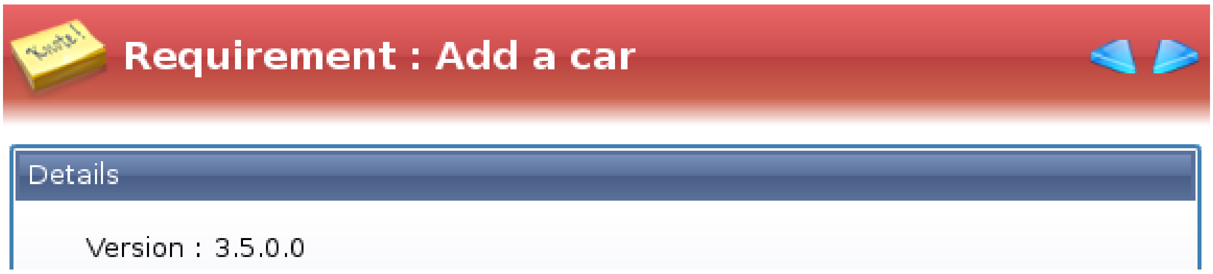


If the changes are not related to a change of need (eg : correcting a misspelling in the name or the description, modify the level of priority...etc) click on [**Keep version**]. In this case, modifications are saved normaly.

If the changes are related to a change of need, click on [**Upgrade**].

Take for example the requirement «Length of a car name can't be greater than 50 caracters». This requirement exists in version 3.4.0.0 of the Rent Software. In the version 3.5.0.0, a new need indicates that «Length of a car name can't be greater than 100 caracters». So, it's an evolution of the need related to the requirement. In this case, the upgrade permits to create a new version and keep track of the previous version.

The requirement history (previous versions and next versions) is available from left and right arrows located in the top-right corner of the screen entry.



Move a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

Select the requirement and move it keeping mouse button pressed. Release the mouse button onto desired destination.

P.S : Several requirements can be moved simultaneously to the same destination by using the [**Ctrl**] and [**Maj**] keys.

Copy a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

Select the requirement, press and maintain pressed the [**Ctrl**] key, then move the requirement onto desired destination.

Release the [Ctrl] key after having released the left mouse button.

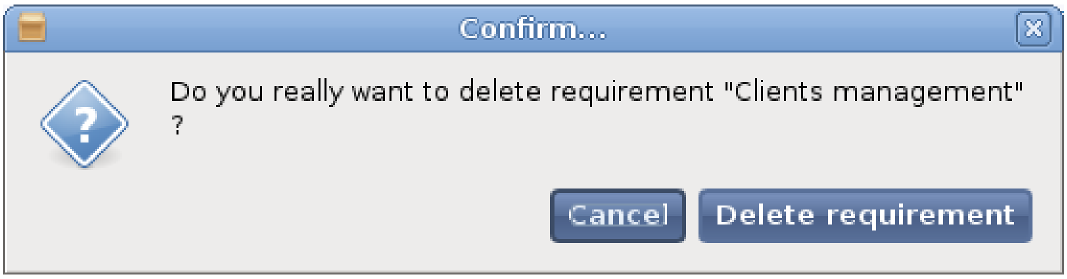
P.S : Several requirements can be copied simultaneously to the same destination by using the [**Ctrl**] and [**Maj**] keys.

Delete a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

Select the requirement from the left tree then select the menu [**Requirements**][**Delete selected requirement**] or right click on the requirement and select [**Delete selected requirement**] from the contextual menu.

The following confirmation message is displayed :



Click on [**Delete requirement**] to validate deletion else click on [**Cancel**].

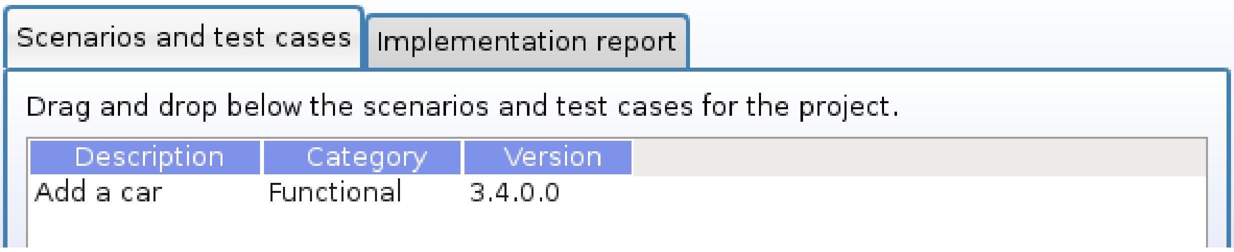
Link scenarios and test cases to a requirement

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

The association between requirements and scenarios permits to know the requirements coverage on a software, according to results of tests executions.

This notion is therefore essential to ensure the completeness of tests on software.

To associate scenarios to a requirement, select the requirement in the left tree of the project.



The screen entry of the requirement is displayed.

Select one or more scenarios in the right tree of the project by holding down the left mouse button and move those scenarios to the area bellow the tab **Scenarios and test cases** into the screen entry of the requirement :

Click on [**Save**] to validate the association between dropped scenarios and the currently edited requirement.

Double-click on to the line corresponding to a scenario in order to display the screen entry of the scenario. Use navigation buttons to come back to the screen entry of the requirement (see Navigation).

To delete association between a scenario and a requirement, select scenarios to delete from the tab **Scenarios and test cases** and press the [**Del**] key, then click on [**Save**] to validate changes.

Identify associated scenarios/test cases

Prerequisites : Have opened a project being connected with a user having access rights at least for reading on requirements and on scenarios of the project (see Permissions management).

To identify scenarios and test cases associated with severals requirements, select one or more requirements in the left tree of the project, then select the menu [**Requirements**][**Select dependant tests**], or click right on a requirement and select [**Select dependant tests**] from the contextual menu.

Dependant scenarios and test cases will be automaticaly selected into the right tree of the project.

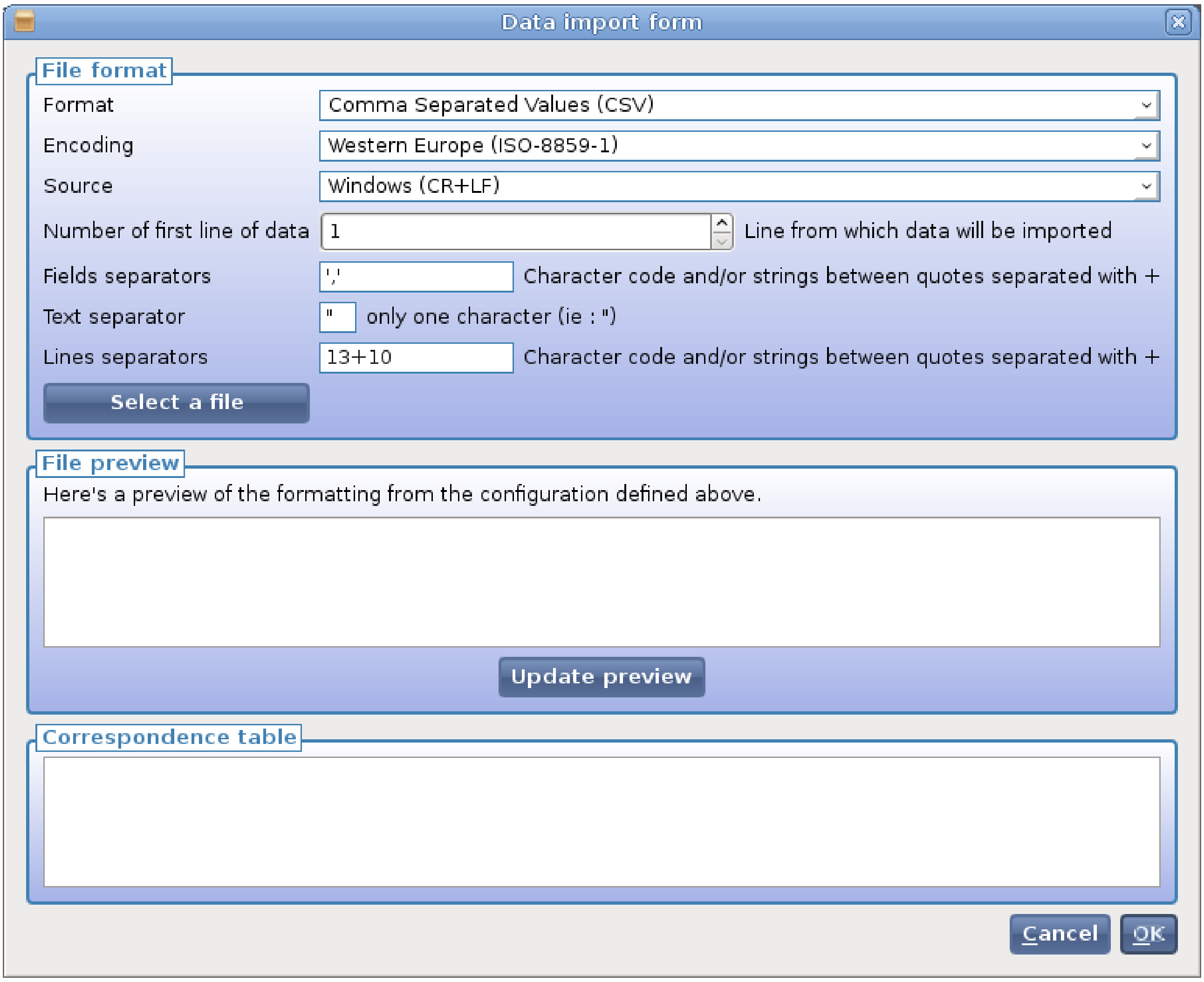
Import requirements from a text file

Prerequisites : Have opened a project being connected with a user having access rights for writing on project requirements (see Permissions management).

This feature permits to import a set of requirements from a text file (ex : CSV) where each line is associated with a requirement.

From the main project window, select the menu [**Requirements**][**Import requirements**] or right click on the left tree of the project and select [**Import requirements**] from the contextual menu.

The following window is displayed :



The section **File format** permits indicate the format of the file to import:

- **Format** : that is the format of the field separator. Following values are available:
    - comma separated values (csv)
    - values separated by semicolons
    - tab separated values
    - custom: indicate custom values into **Field separator**
- **Encoding** : that is the kind of caracters. Use encoding to resolve problems of accentued caracters. Available values are :
    - Western europe (ISO-8859-1)
    - Unicode (UTF-8)
- **Source** : that is the information for record separator. Use the field **Line separator** bellow to customize this value**.** Available values are :
    - Windows (CR+LF)
    - Other (CR)
- **Number of first line of data** : indicate from which line the file will be read
- **Field separator** : permits the customization of the field separator. This value is update according to **Format**.
- **Text separator** : permits the customization of the text separator. This separator is used to identify the start and the end of a field. This is usefull when some fields contain carriage return.
- **Line separator** : permits the customization of the record separator. This value is update according to **Source :**
    - for Windows (CR+LF) this is 10+13
    - for Other (CR) this is 10

Click on [**Select a file**] to load the source file.

The file preview is displayed.

If you modify the file format, click on [**Update preview**] to show the new preview.

Correspondence table is used to associate data from the source file to data used in R.T.M.R requirements. For each column read, you can indicate if :

- it has **No correspondence**
- it is the **Requirement name** : if anti-slash (caracter \) are read in the column, they are interpreted as delimiter of hierarchical levels.
- It is the **Requirement description**

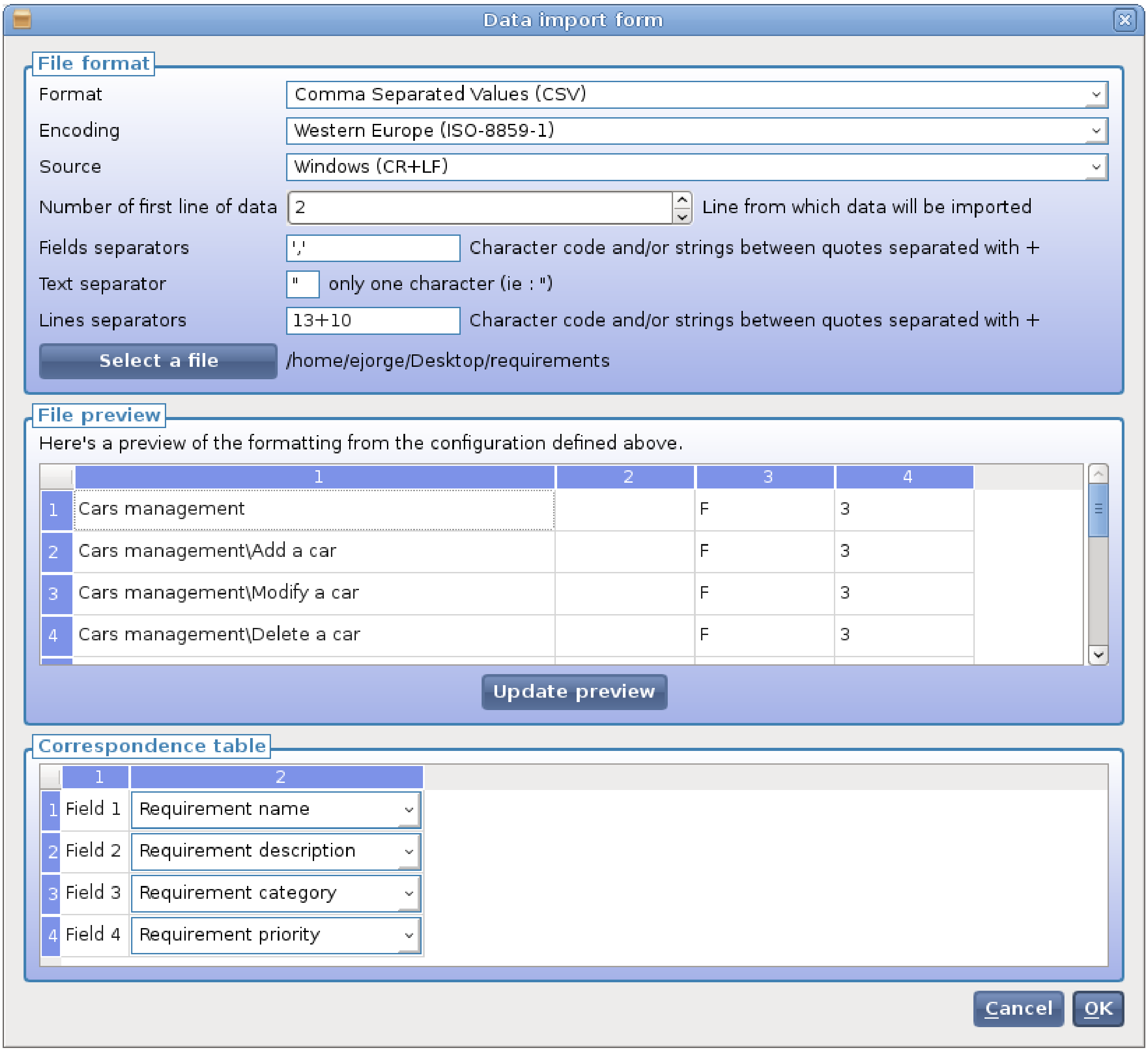
- it is the **Requirement category**. Known values are :
  - F for Fonctionnal
  - I for Interoperability
  - C for Load
  - P for Performance
  - D for Disponibility
  - S for Security
  - E for Exploitability
  - Z for Other
- it is the **Requirement priority**. Values must be between 1 and 5.

Then, click on [**OK**] import all data.

Below a sample CSV file containing several hiearchical levels :

```
"Name","Description","Category","Priority"
"Cars management","","F","3"
"Cars management\Add a car","","F","3"
"Cars management\Modify a car","","F","3"
"Cars management\Delete a car","","F","3"
"Assurances management","","F","3"
"Assurances management\Add an assurance","","F","3"
"Assurances management\Modify an assurance","","F","3"
"Assurances management\Delete an assurance","","F","3"
"Clients management","","F","3"
"Clients management\Add a client","","F","3"
"Clients management\Modify a client","","F","3"
"Clients management\Delete a client","","F","3"
[...]
```

Below see the corresponding preview :



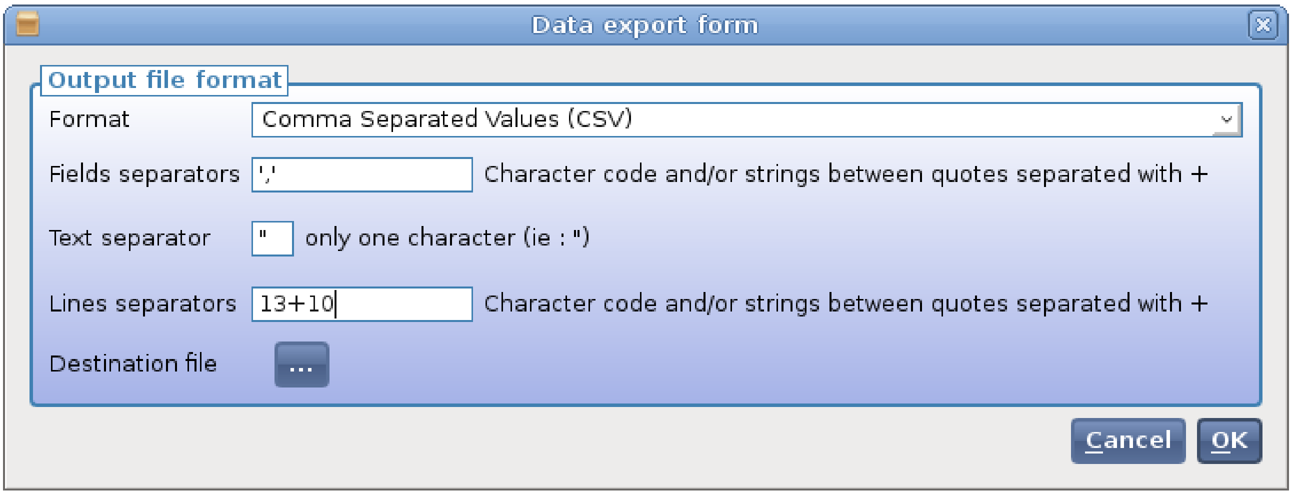
Export requirements to a text file

Prerequisites : Have opened a project being connected with a user having access rights for reading on project requirements (see Permissions management).

This feature permits export a set of requirements to a text file (ex : CSV).

Select all the requirements to export from the left tree of the project, then select the menu [**Requirements**][**Export requirements**] or right click on the left tree of the project and select [**Export requirements**] from the contextual menu.

The following screen appears :



The section **Output file format** indicates the format of the file to generate :

- **Format** : this is the field separator. Available values are :
  - comma separated value (csv)
  - values separated by semicolons
  - tab separated values
  - custom: indicate custom values into **Field separator**
- **Field separator** : permits the customization of the field separator. This value is updated according to **Format**.
- **Text separator** : permits the customization of the text separator. This separator is used to identify the start and the end of a field. This is usefull when some fields contain carriage return.
- **Line separato**r : permits the customization of the separator between each exported requirement.

Click on [**...**] to select the destination file.

Then, click on [**OK**] to export data.

P.S : Selected requirements and its children will all be exported.

The anti-slash caracter (\) will be used to delimitate hierarchical levels between requirements and its children.

Below see a sample of an exported file (field separator : tab, text separator : none):

Name	Description	Category	Priority
Cars management		F	3
Cars management\Add a car		F	3
Cars management\Modify a car		F	3
Cars management\Delete a car		F	3
Assurances management		F	3

Assurances management\Add an assurance	F	3
Assurances management\Modify an assurance	F	3
Assurances management\Delete an assurance	F	3
Clients management	F	3
Clients management\Add a client	F	3
Clients management\Modify a client	F	3
Clients management\Delete a client	F	3

Management of scenarios and test cases

Introduction

In order to structure the management of tests, the test repository is presented in a hierarchical list. The interest is to create sets of tests, group by fonctional module, or by test type, or by priority...etc. Each test or test group may contain subgroups of tests called children tests.

The term **test** is associated with either a **scenario**, or **a test case**.
The representation of a scenario and the representation of a test case is the same: it is a node in the tests tree.

The difference lies in the content :

- **a test case** contains a series of actions to validate or invalidate the outcome of a function from the tested software : it should not contain children.
- **a scenario** can be defined as a set of test cases : therefore it contains children but should not contain actions

A good approach is to group sets of scenarios by center of interest.

Reusable tests

In order to factor out all the tests, the concept of reusable test was implemented.

Its instanciation is done through the ability to create :

- linked scenarios
- call for test cases.

Imagine following scenarios and test cases into a software managing cars rent :

- Test case : « Connection as an administrator»
  - Action : enter login and password
  - Action : Click on [OK] button
- Scenario : «Add a car»
  - Test case : « Connection as an administrator»
    - Action : enter login and password
    - Action : Click on [OK] button
  - Test case : «Find and add a car»
    - Action : enter car registration number
    - Action : Click on find button
    - Action : if the car is not found click on [Add] in the search screen
- Scenario : « Modify a car »

- Test case « Connection as an administrator»
  - Action : enter login and password
  - Action : Click on [OK] button
- Test case «Find and modify a car»
  - Action : enter car registration number
  - Action : Click on find button
  - Action : Click onto the link associated to the search car

- Scenario : «Remove a car»
  - Test case « Connection as an administrator»
    - Action : enter login and password
    - Action : Click on [OK] button
  - Test case «Find and remove a car»
    - Action : enter car registration number
    - Action : Click on find button
    - Action : Click on [Delete] button located front to the desired car
- Test case : «Find a car»
  - Action : enter car registration number
  - Action : Click on find button

<u>Linked scenarios :</u>

Scenarios «Add a car», « Modify a car » and «Remove a car» depend on the test case « Connection as an administrator» as to add, modify or remove a car in the software, user must be connected as an administrator. In this case we could add a child scenario for all of three scenarios that will link to the test case named « Connection as an administrator».

Thus, any change in the test case « Connection as an administrator» in future versions of the software permits to easily identify the regression tests to be performed on dependant scenarios like «Add a car», « Modify a car » and «Remove a car».

P.S : for more information to know ho to create a linked scenario see Create a link to a scenario.

<u>Call of test cases :</u>

The test case «Find a car» described how to search a car from its registration number.

This test case can be reused in «Add a car», «Modify a car» and «Remove a car».

Indeed, «Find and add a car», «Find and modify a car» and «Find and remove a car» implies similar actions.

In this case, we could do a call to a generic test named «Find a car» in children test cases of scenarios «Add a car», «Modify a car» and «Remove a car».

P.S : for more information to know ho to create a call to a test case see Create a call to a test case.

Add a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

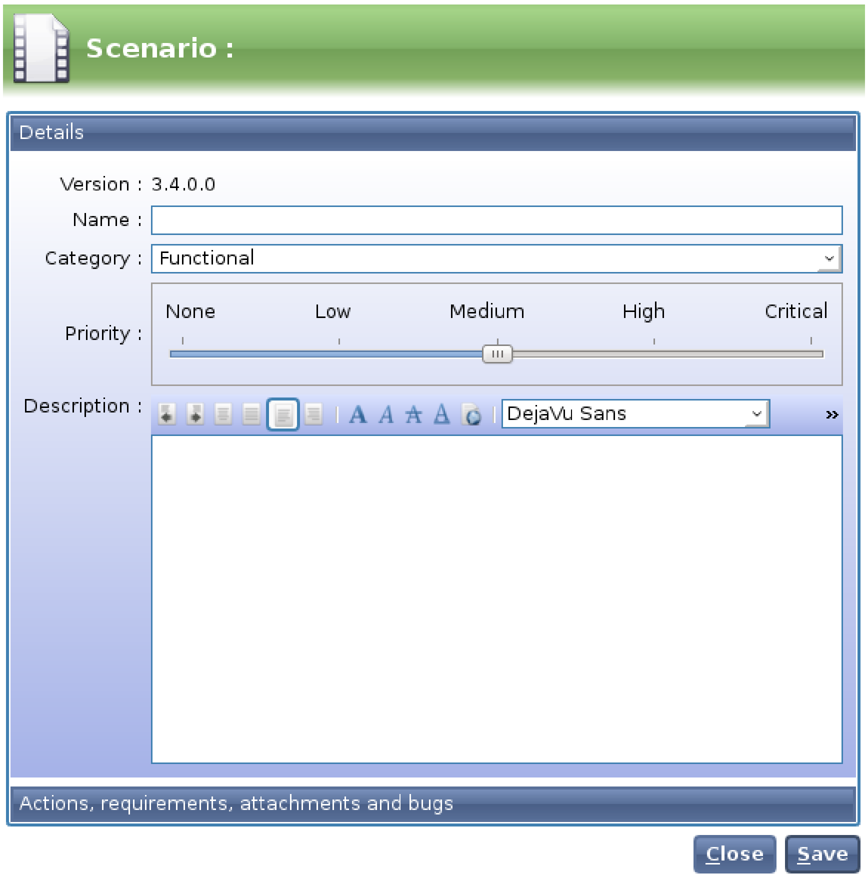
From the main window of the open project, select the menu [**Tests**][**New test**] or do a right click in the right side panel of the scenarios and test cases and select [**New test**].

By default, the test is created directly in the tree and is positioned after the selected test when handling.

To create a child test, select the menu [**Tests**][**Add child test**] or do a right click in the right side panel of the scenarios and test cases and select [**Add child test**].

By default, the child test is directly created in the tree and is positioned below the selected test when handling.

The input screen of the new test is displayed in the main window of the project :



The version of the test is set automatically: it is the same version as the open project.

It remains only to enter information about the test :

- **Name** : mandatory (eg : Manage cars)
- **Category** : this is the type of the test or the test group :
  - Fonctionnal (eg : manage cars of any mark)
  - Interoperability (eg : import car files)

- Load : (eg : handle 50 simultaneous users)
- Performance (eg : search result of the customer database in less than 10 seconds)
- Disponibility (eg : handle connection all the year, 24/24 and 7/7)
- Security (eg : restricted access to super-user for insurers data)
- Exploitability (eg : generate daily activity reports for rental)
- Other (eg : support accessibility for disabled users)

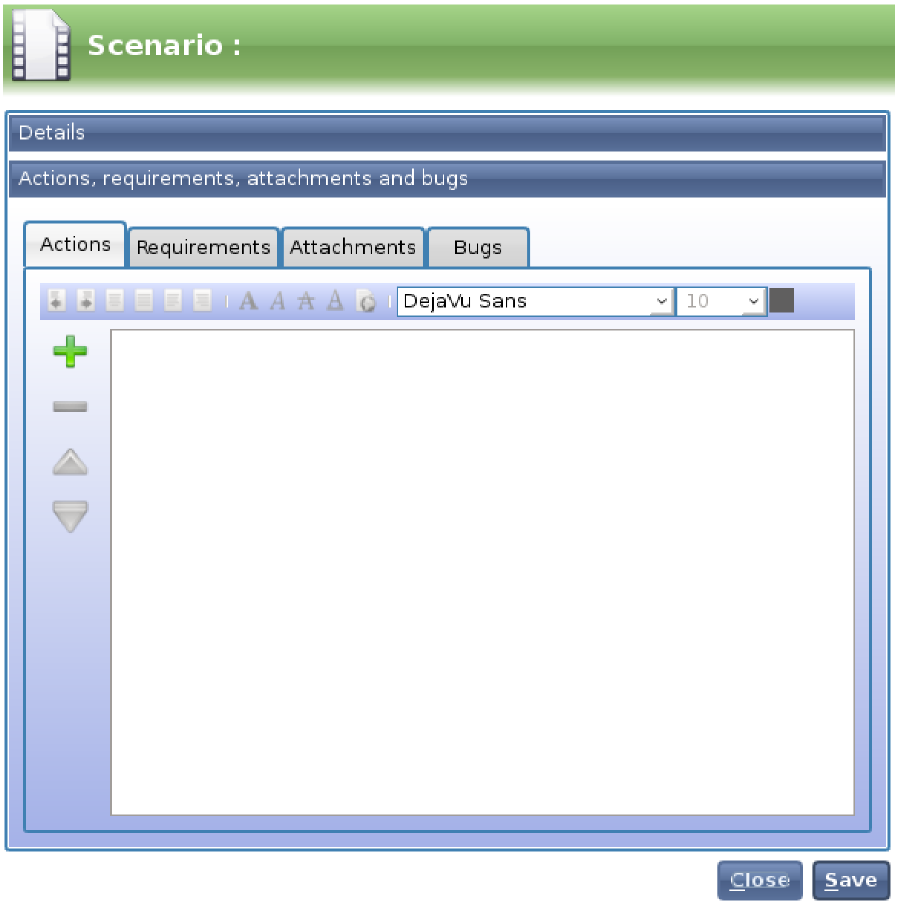
- **Description** : this is the description of the test
- **Priority** : that's the test matter level
  - critical
  - high
  - medium
  - low
  - none

If the test is a scenario, ie a group of sub-scenarios or group of test cases, click on [**Save**] to validate entered data.

If the test is a test case, ie a set of actions, enter actions as outlined below.

Manage actions of a test case

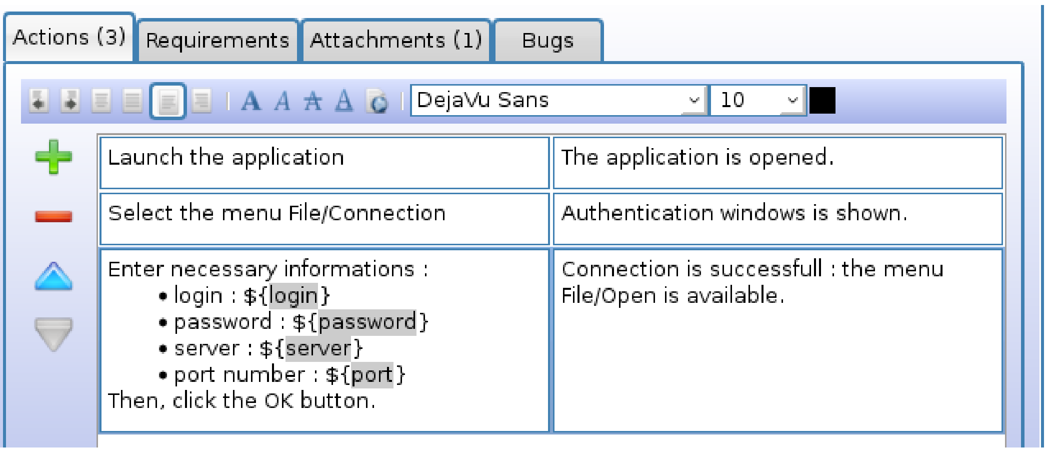
To enter action click on the section [**Actions, requirements, attachments and bugs**].



Use buttons [+], [-] and arrows [**top**] and [**bottom**] to modify actions list :

- button [+] : add an action after the selected action
- button [-] : delete the selected action
- arrow [top] : move up the selected action
- arrow [**bottom**] : move down the selected action

Use the toolbar layout to enhance the format of the input text.



The left column contains the description of the action to achieve. The column on the right, contains the description of the outcome of the action.

In the example above some parameters are used (${server}, ${port}, ${login}, ${password}).

These parameters are evaluated at the time of scenario execution as part of a test campaign.

For more information on execution parameters refer to Manage project parameters.

Parameters are usable in the description field and in the outcome field of actions and into the description field of the edited test.

To use parameters, simply type the beginning of the name of a parameter and a list will automatically displays all the parameters corresponding to the entered name : this is auto-completion mode.

Otherwise by pressing the keys [**Ctrl**] and [**Espace**] the complete list of project parameters is displayed.

P.S : if a parameter is added while editing a test, you must reload the form of the test to have access for the new parameter, if you don't want to reload the test enter directly the paramter name surrounded by characters $, { and }.

Manage requirements

To associate requirements to scenarios / tests cases, refer to Identify requirements associated to a scenario / test case.

Manage attachements

Select tab **Attachments** from the test screen.

From the file browser provided by the system, drag and drop one or more files into the area below the tab and click on [**Save**] to confirm the addition of attachments.

Manage bugs

The tab **Bugs** from the test screen allows only to consult of associated bugs.

Entering bugs is done when running a tests campaign (cd Add a bug when executing a test).

Create a set of tests from a set of requirements

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests and for reading on project requirements (see Permissions management).

To facilitate the establishment of the scenarios and test cases repository, a simple drag and drop of one or more requirements fro mthe requirements tree to the tests tree, create all corresponding tests.

Select a requirement from the requirements tree and move it by holding down the left mouse button to the tests tree. Release the button on the desired destination.

P.S : Several requirements can be copied simultaneously to the same destination by using the [**Ctrl**] and [**Maj**] key when selecting requirements.

The tests created in this way will automatically be associated to the requirements.

See : Identify requirements associated to a scenario / test case.

Modify a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

Left click on the test to modify in the pannel on the right of the main project window.

The entry screen of the test is displayed (voir Add a scenario / test case).

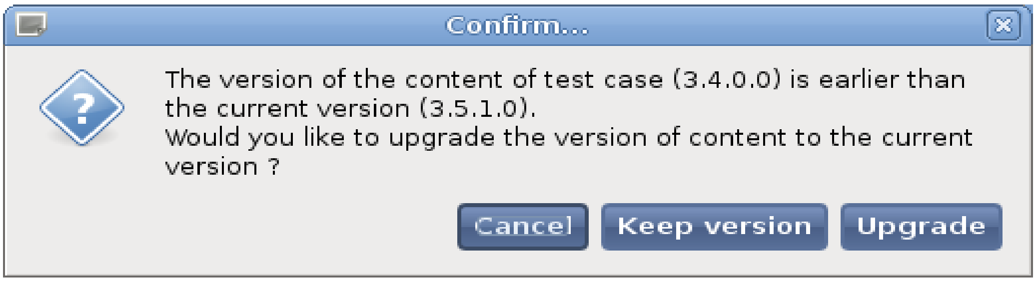
Modify requirement data then click on [**Save**].

Scenario 1 - test version is the same as project version :

In this case, changes are saved.

Scénario 2 - test version is different from project version :

In this case, the following message is displayed :

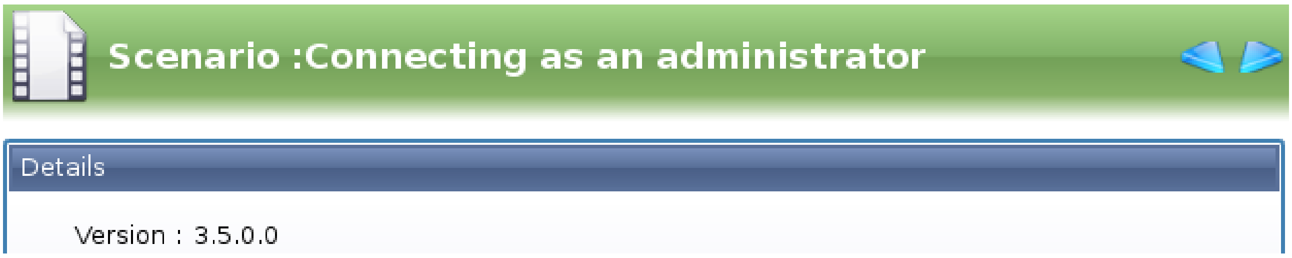


If the changes are not related to a change of need (eg : correcting a misspelling in the name or the description, modify the level of priority...etc) click on [**Keep version**]. In this case, modifications are saved normaly.

If the changes are related to a change of need, click on [**Upgrade**].

It may be the case when actions are added (eg : entering new field in a form).

The test history (previous versions and next versions) is available from left and right arrows located in the top-right corner of the screen entry.



Move a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

Select the test and move it keeping mouse button pressed. Release the mouse button onto desired destination.

P.S : Several tests can be moved simultaneously to the same destination by using the [**Ctrl**] and [**Maj**] keys.

Copy a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

Select the test, press and maintain pressed the [**Ctrl**] key, then move the test onto desired destination.

Release the [Ctrl] key after having released the left mouse button.

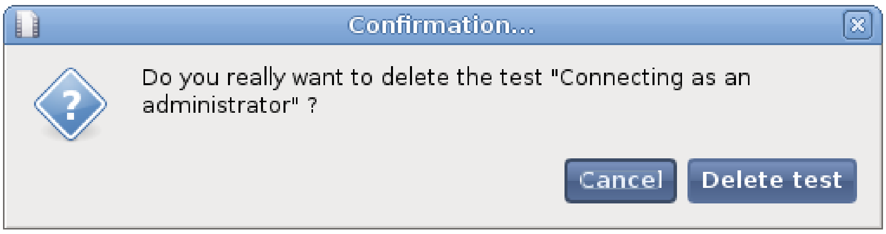
P.S : Several tests can be copied simultaneously to the same destination by using the [**Ctrl**] and [**Maj**] keys.

Delete a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

Select the test from the left tree then select the menu [**Tests**][**Delete selected test**] or right click on the test and select [**Delete selected test**] from the contextual menu.

The following confirmation message is displayed :



Click on [**Delete test**] to validate deletion else click on [**Cancel**].

Identify requirements associated to a scenario / test case

Prerequisites : Have opened a project being connected with a user having access rights at least for reading on project tests and on project requirements (see Permissions management).

The association between requirements and scenarios permits to know the requirements coverage on a software, according to results of tests executions.

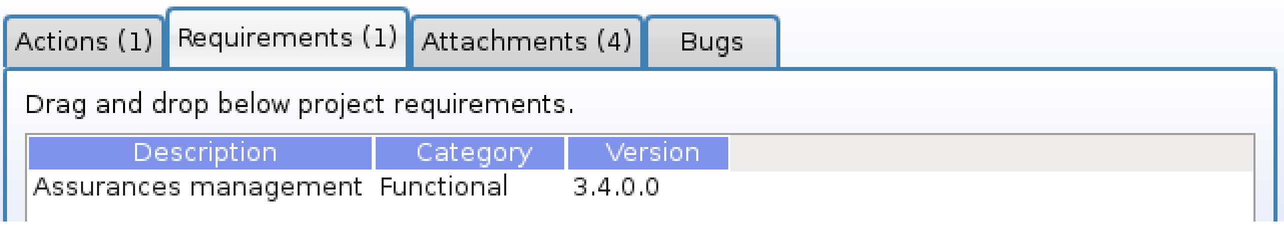
This notion is therefore essential to ensure the completeness of tests on software.

To associate requirements to a test, select the test in the left tree of the project.

The screen entry of the test is displayed.

Select tab **Requirements**.

Select one or more requirements in the left tree of the project by holding down the left mouse button and move those requirements to the area bellow the tab **Requirements** into the screen entry of the test :



Click on [**Save**] to validate the association between dropped requirements and the currently edited test.

Double-click on to the line corresponding to a requirement in order to display the screen entry of the requirement. Use navigation buttons to come back to the screen entry of the test (see Navigation).

To delete association between a scenario and a requirement, select requirements to delete from the **Requirements** tab and press the [**Del**] key, then click on [**Save**] to validate changes.

Create a link to a scenario

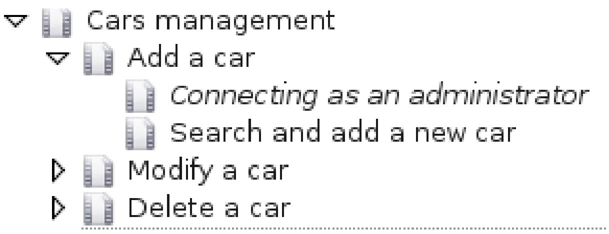
Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

For more information on linked test refer to the section Reusable tests.

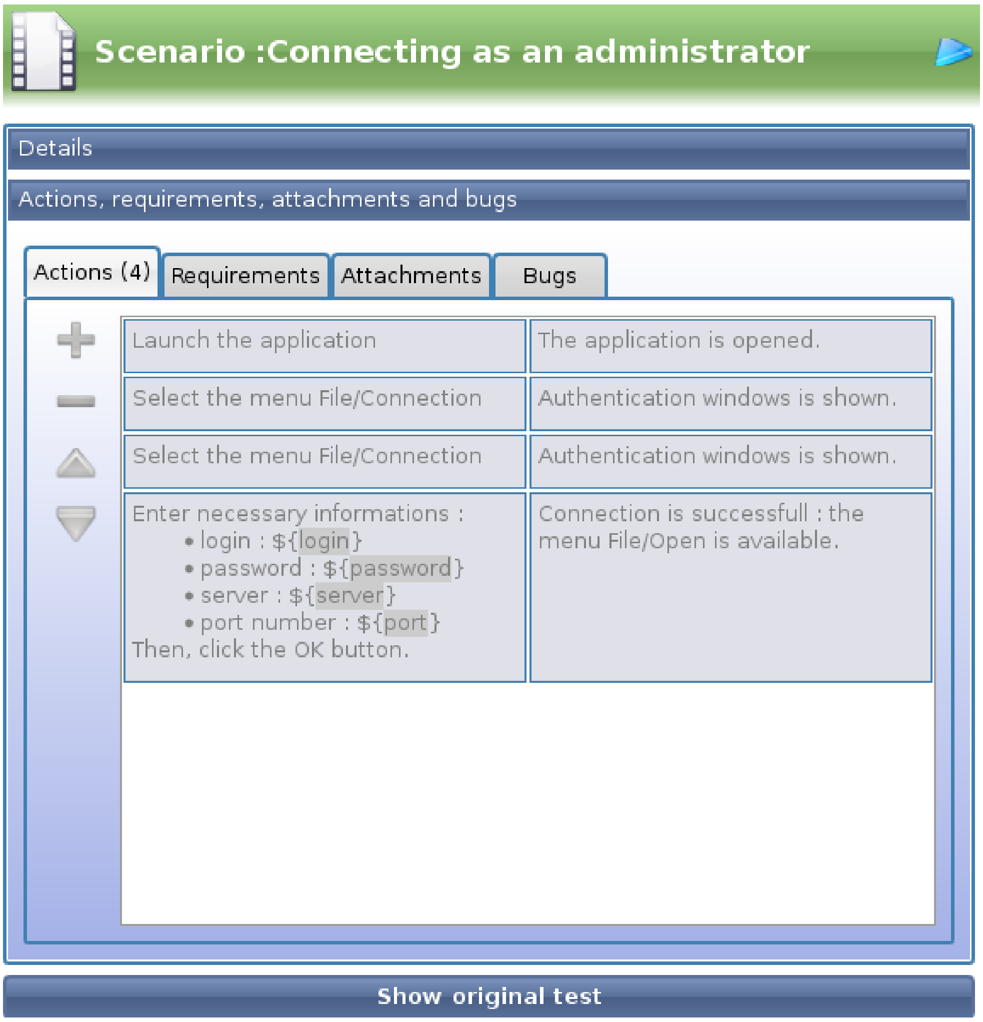
To create a linked test, select the test, press and hold down [**Ctrl**] and [**Shift**] key, then move the test in the tree by pressing the left mouse button.

Release the [Ctrl] and [Shift] key after release the left mouse.

The linked test appears formatted in italic in the tests tree :



The selection of a linked test in the test tree, displays the following screen :



Modify a linked test is not possible, only the original test is editable.
To display original test, click on [**Show original test**].

To display all linked tests from an original test see Select tests linked to an original test.

Create a call to a test case

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

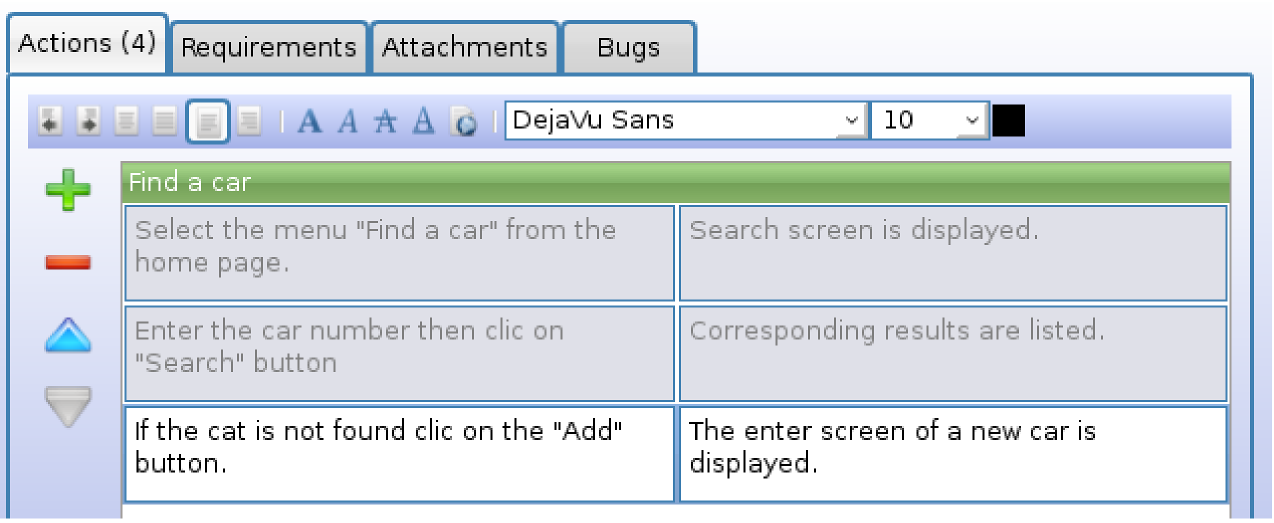
For more information on test call see Reusable tests.

To call a test case, select a test case in the tree located at the right side panel of the project.

The entrer screen of the test case is displayed.

To create a call to another test case, select the test case in the source tree and move it by holding down the left mouse button to the actions list of the test case currently edited.

When a test case uses a different test cases, the list of actions is presented in the following format :



In this example, the currently edited test is calling test case named «Find a car» containing 2 actions (line 2 and 3).

To move the called test, into the actions list, select the green action (line 1) then use arrows [**Up**] and [**Down**] to move it.

If the test case «Find a car» is updated (eg : an action is added) modifications will be dynamicaly reported to calling tests.

Select tests linked to an original test

Prerequisites : Have opened a project being connected with a user having access rights for reading on project tests (see Permissions management).

To identify what are the scenarios and test cases linked to an original test, select a test in the tests tree, then select the menu [**Tests**][**Select dependant tests**], or right click on the test and select [**Select dependant tests**] from the contextual menu.

Parent linked scenarios and test cases will be selected in the tests tree on the right side panel of the project.

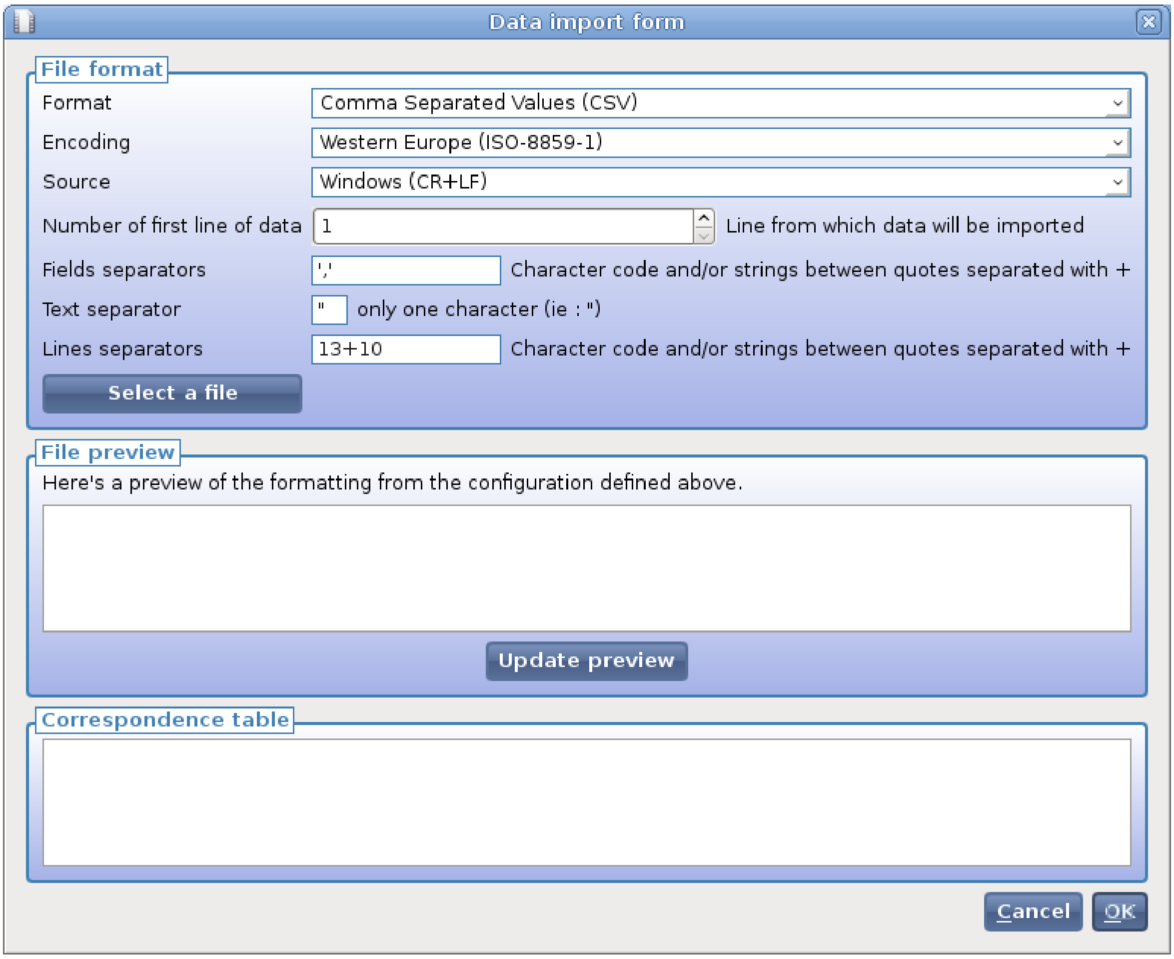
Import tests from a text file

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

This feature permits to import a set of scenarios and test cases from a text file (ex : CSV) where each line is associated with a scenario and / or an action of a test case.

From the main project window, select the menu [**Tests**][**Import tests**] or right click on the right tree of the project and select [**Import tests**] from the contextual menu.

The following window is displayed :



The section **File format** permits indicate the format of the file to import:

- **Format** : that is the format of the field separator. Following values are available:
  - comma separated values (csv)
  - values separated by semicolons
  - tab separated values
  - custom: indicate custom values into **Field separator**

- **Encoding** : that is the kind of caracters. Use encoding to resolve problems of accentued caracters. Available values are :
  - Western europe (ISO-8859-1)
  - Unicode (UTF-8)
- **Source** : that is the information for record separator. Use the field **Line separator** bellow to customize this value**.** Available values are :
  - Windows (CR+LF)
  - Other (CR)
- **Number of first line of data** : indicate from which line the file will be read
- **Field separator** : permits the customization of the field separator. This value is update according to **Format**.
- **Text separator** : permits the customization of the text separator. This separator is used to identify the start and the end of a field. This is usefull when some fields contain carriage return.
- **Line separator** : permits the customization of the record separator. This value is update according to **Source :**
  - for Windows (CR+LF) this is 10+13
  - for Other (CR) this is 10

Click on [**Select a file**] to load the source file.

The file preview is displayed.

If you modify the file format, click on [**Update preview**] to show the new preview.

Correspondence table is used to associate data from the source file to data used in R.T.M.R tests. For each column read, you can indicate if :

- it has **No correspondence**
- it is the **Requirement name** : if anti-slash (caracter \) are read in the column, they are interpreted as delimiter of hierarchical levels.
- It is the **Test description**
- it is the **Test category**. Known values are :
  - F for Fonctionnal
  - I for Interoperability
  - C for Load
  - P for Performance
  - D for Disponibility

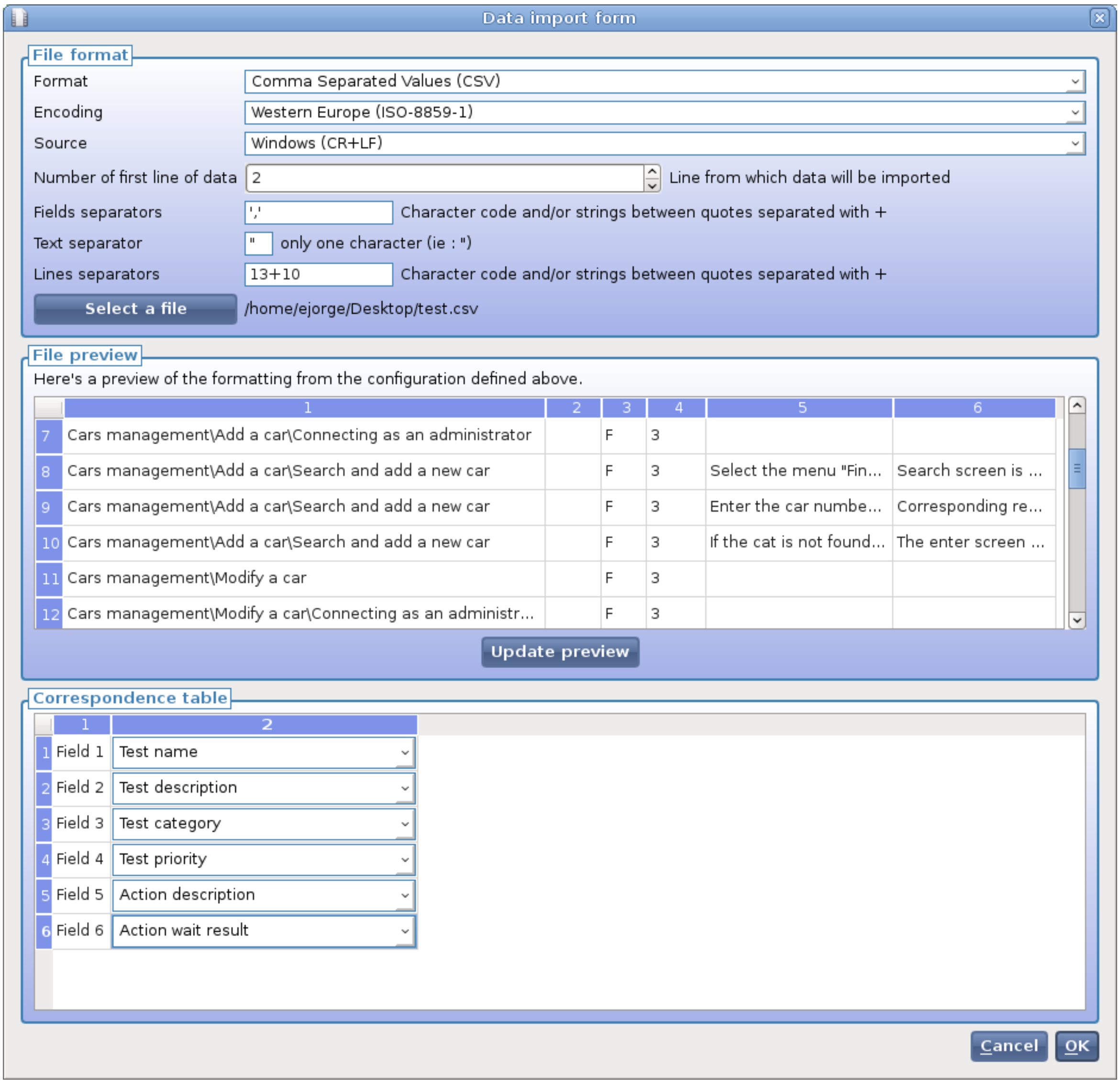
   - S for Security
   - E for Exploitability
   - Z for Other
- it is the **Test priority**. Values must be between 1 and 5.

Then, click on [**OK**] import all data.

Below a sample CSV file containing several hiearchical levels :

```
"Name","Description","Category","Priority","Action","Wait result"
"Connecting as an administrator","","F","3","Launch the application","The application is opened."
"Connecting as an administrator","","F","3","Select the menu File/Connection","Authentication windows is shown."
"Connecting as an administrator","","F","3","Enter necessary informations :
login : ${login}
password : ${password}
server : ${server}
port number : ${port}
Then, click the OK button.","Connection is successfull : the menu File/Open is available."
"Cars management","","F","5"
"Cars management\Add a car","","F","3"
"Cars management\Add a car\Connecting as an administrator","","F","3"
"Cars management\Add a car\Search and add a new car","","F","3","Select the menu ""Find a car"" from the home page.","Search screen is displayed."
"Cars management\Add a car\Search and add a new car","","F","3","Enter the car number then clic on ""Search"" button","Corresponding results are listed."
"Cars management\Add a car\Search and add a new car","","F","3","If the cat is not found clic on the ""Add"" button.","The enter screen of a new car is displayed."
"Cars management\Modify a car","","F","3"
"Cars management\Modify a car\Connecting as an administrator","","F","3"
"Cars management\Modify a car\Search and modify a car","","F","3"," □," □
"Cars management\Delete a car","","F","3"
"Cars management\Delete a car\Connecting as an administrator","","F","3"
"Cars management\Delete a car\Search and delete a car","","F","3","□",""
"Find a car","","F","3","Select the menu ""Find a car"" from the home page.","Search screen is displayed."
"Find a car","","F","3","Enter the car number then clic on ""Search"" button","Corresponding results are listed."
"Assurances management","","F","3"
"Assurances management\Add an assurance","","F","3"
"Assurances management\Add an assurance\Connecting as an administrator","","F","3"
"Assurances management\Modify an assurance","","F","3"
"Assurances management\Modify an assurance\Connecting as an administrator","","F","3"
"Assurances management\Delete an assurance","","F","3"
"Assurances management\Delete an assurance\Connecting as an administrator","","F","3"
"Clients management","","F","3"
"Clients management\Add a client","","F","3"
"Clients management\Add a client\Connecting as an administrator","","F","3"
"Clients management\Modify a client","","F","3"
"Clients management\Modify a client\Connecting as an administrator","","F","3"
"Clients management\Delete a client","","F","3"
"Clients management\Delete a client\Connecting as an administrator","","F","3"
```

Below see the corresponding preview :



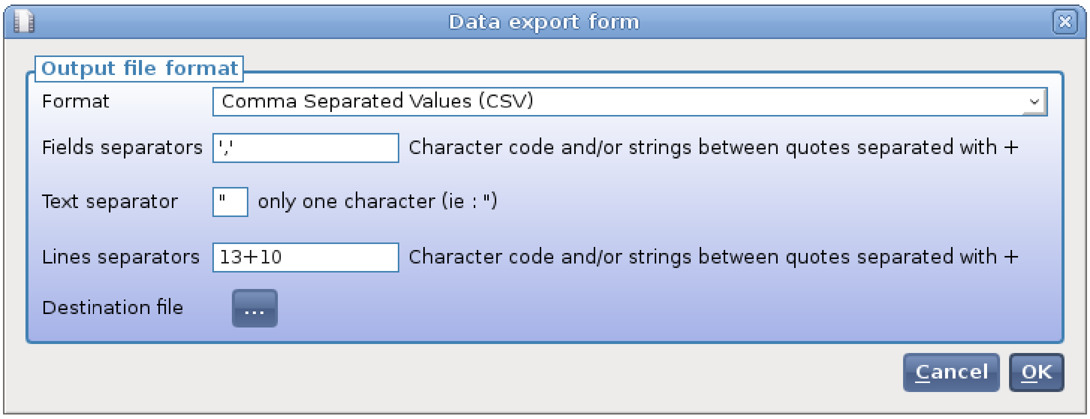
Export tests to a text file

Prerequisites : Have opened a project being connected with a user having access rights for writing on project tests (see Permissions management).

This feature permits export a set of scenarios and/or test cases to a text file (ex : CSV).

Select all the tests to export from the irght tree of the project, then select the menu [**Tests**][**Export tests**] or right click on the right tree of the project and select [**Export tests**] from the contextual menu.

The following screen appears :



The section **Output file format** indicates the format of the file to generate :

- **Format** : this is the field separator. Available values are :
    - comma separated value (csv)
    - values separated by semicolons
    - tab separated values
    - custom: indicate custom values into **Field separator**
- **Field separator** : permits the customization of the field separator. This value is updated according to **Format**.
- **Text separator** : permits the customization of the text separator. This separator is used to identify the start and the end of a field. This is usefull when some fields contain carriage return.
- **Line separato**r : permits the customization of the separator between each exported test.

Click on [**...**] to select the destination file.

Then, click on [**OK**] to export data.

P.S : Selected tests and its children will all be exported.

The anti-slash caracter (\) will be used to delimitate hierarchical levels between tests and its children.

Below see a sample of an exported file (field separator : tab, text separator : none):

```
"Name"	"Description"	"Category"	"Priority"	"Action"	"Wait result"
"Cars management\Add a car"	""	"F"	"3"
"Cars management\Add a car\Connecting as an administrator"	""	"F"	"3"
"Cars management\Add a car\Search and add a new car"	""	"F"	"3"	"Select the menu ""Find a car"" from the home page."	"Search screen is displayed."
"Cars management\Add a car\Search and add a new car"	""	"F"	"3"	"Enter the car number then clic on ""Search"" button"	"Corresponding results are listed."
"Cars management\Add a car\Search and add a new car"	""	"F"	"3"	"If the cat is not found clic on the ""Add"" button."	"The enter screen of a new car is displayed."
```

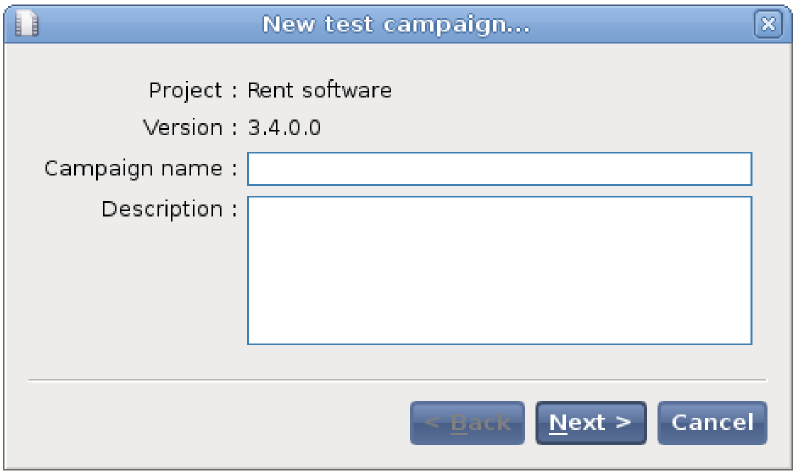
Campaigns management

Create a tests campaign

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

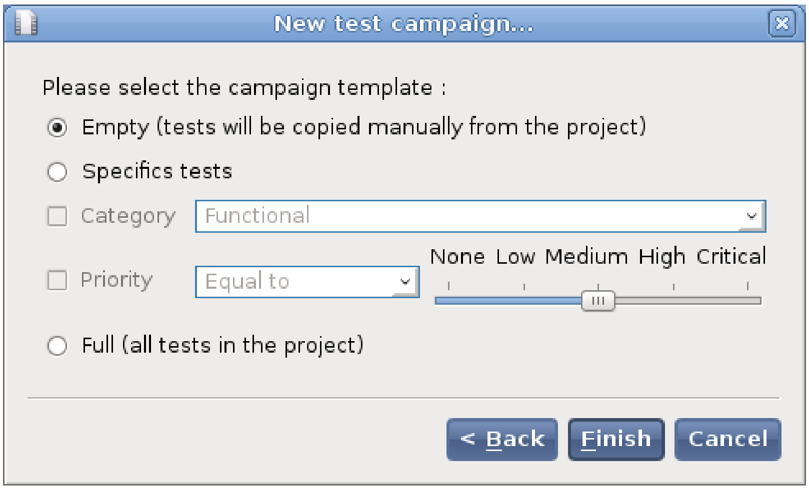
Select the menu [**Campaigns**][**New campaign...**] or right click on to the lower left panel of the main window and select the menu [**New campaign ...]** from the contextual menu.

A wizard is displayed :



Enter the name of the tests campaign (eg : «Car module checking»), its description (eg: «This campaign target all features concerning cars»), then click on [**Next**].

The following screen appears :



Choose a model of campaign :

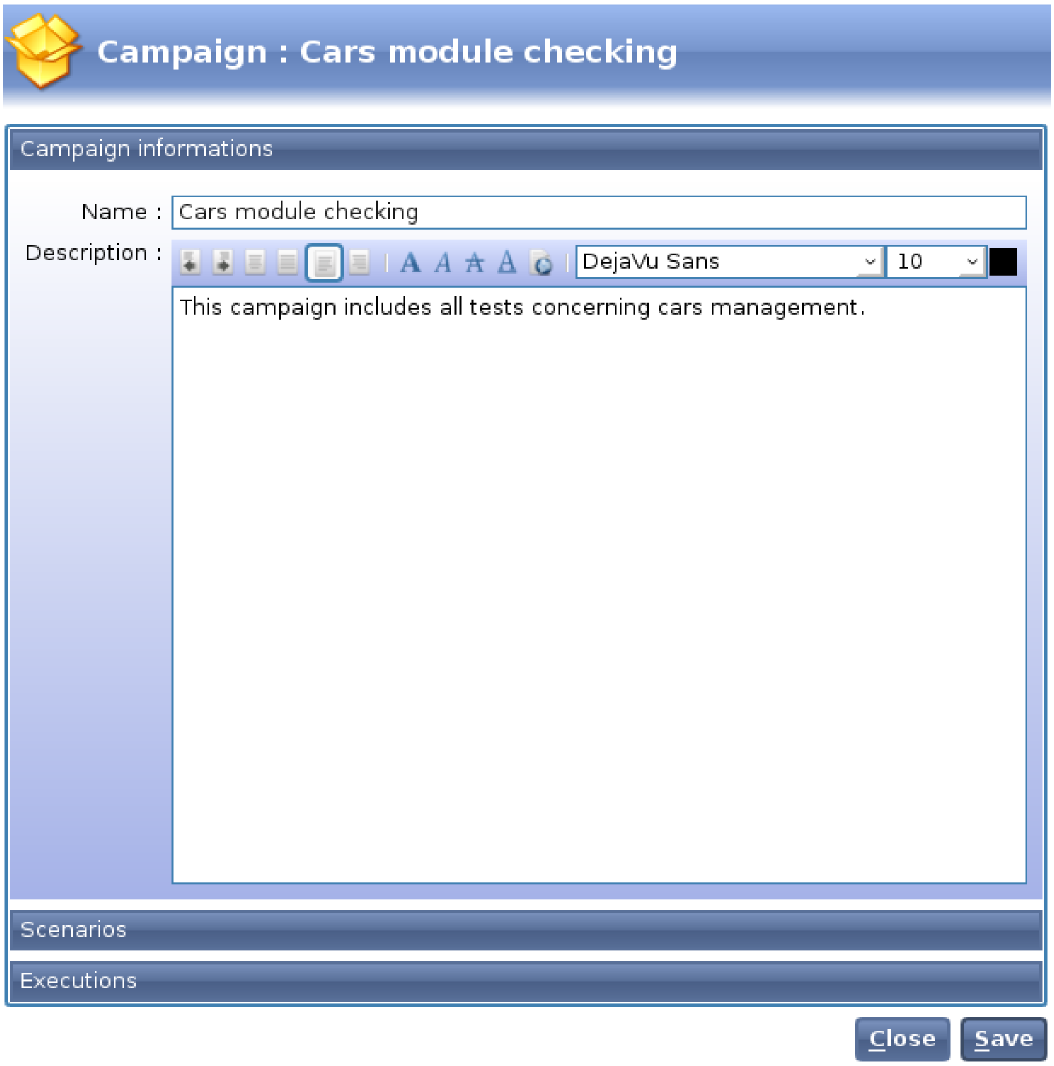
- option «**Empty**» creates a vacuum test campaign that will be filled manually by selecting requirements and tests in their respective tree.
- option «**Specifics tests**» can pre-select a set of tests from :
  - their category (fonctionnal, load, performance, interoperability...etc)

– their priority

– option «**Full**» pre-select all tests of the current project version.

Select desired campaign model then click on [**Finish**]. This button is inactive if the name of the campaign has not been entered on the first screen of the wizard.

Then the following window is displayed :



Add tests from scenarios of the project

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

Select the campaign from the list located at the lower left corner of the main window.

Then select the **Scenarios** box.

To add tests to the campaign, select the tests in the tree located at the right side panel of the project and drag the tests in the **Scenarios** area of the campaign screen.

P.S : Use [**Ctrl**] and [**Maj**] keys to select several tests from the tests tree.

It is quite possible to insert several times, the same tests in a single campaign.
The goal could be, for example, execute a script with different parameters

Add tests from requirements of the project

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

Select the campaign from the list located at the lower left corner of the main window.

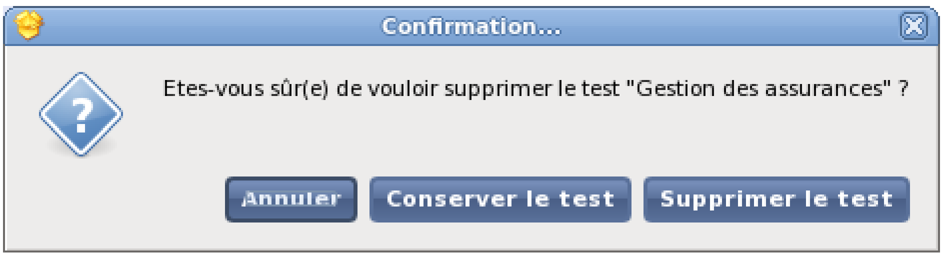
The method of adding tests via requirements is to select the requirements to cover during the campaign.
To do this, select requirements in the left side panel and drag them in the **Scenarios** area of the campaign.
This operation has the effect of automatically selecting the tests associated with selected requirements and insert them into the campaign.
PS: Use [**Ctrl**] and [**Shift**] keys to select the requirements in the tree

Delete tests of a campaign

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

Select the campaign from the list located at the lower left corner of the main window.

To delete tests in a campaign, select the tests in the **Scenarios** area, then press the [**Delete**] key.
A confirmation window will ask you to validate the choice of deletion:



Caution: click [**Delete test**] removes the test campaign.

Modify a tests campaign

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

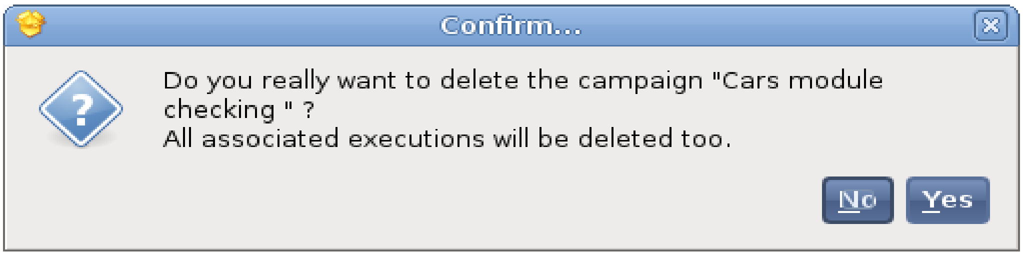
Modification of a test campaign consists to add, delete or move scenarios.
These changes are possible even if the campaign has been executed.
In other words we just run a test campaign, then change a posteriori tests which it contains.

Delete a tests campaign

Prerequisites : Have opened a project being connected with a user having access rights for writing on project campaigns (see Permissions management).

Select the campaign from the list located at the lower left corner of the main window.

Select the menu [**Campaign**] [**Delete selected campaign**] or click with the right mouse button on the campaign in the lower left side panel and select [**Delete selected campaign**].
A confirm window is displayed to confirm your choice :



To permanently delete the campaign, click on [**Yes**], otherwise click [**No**].

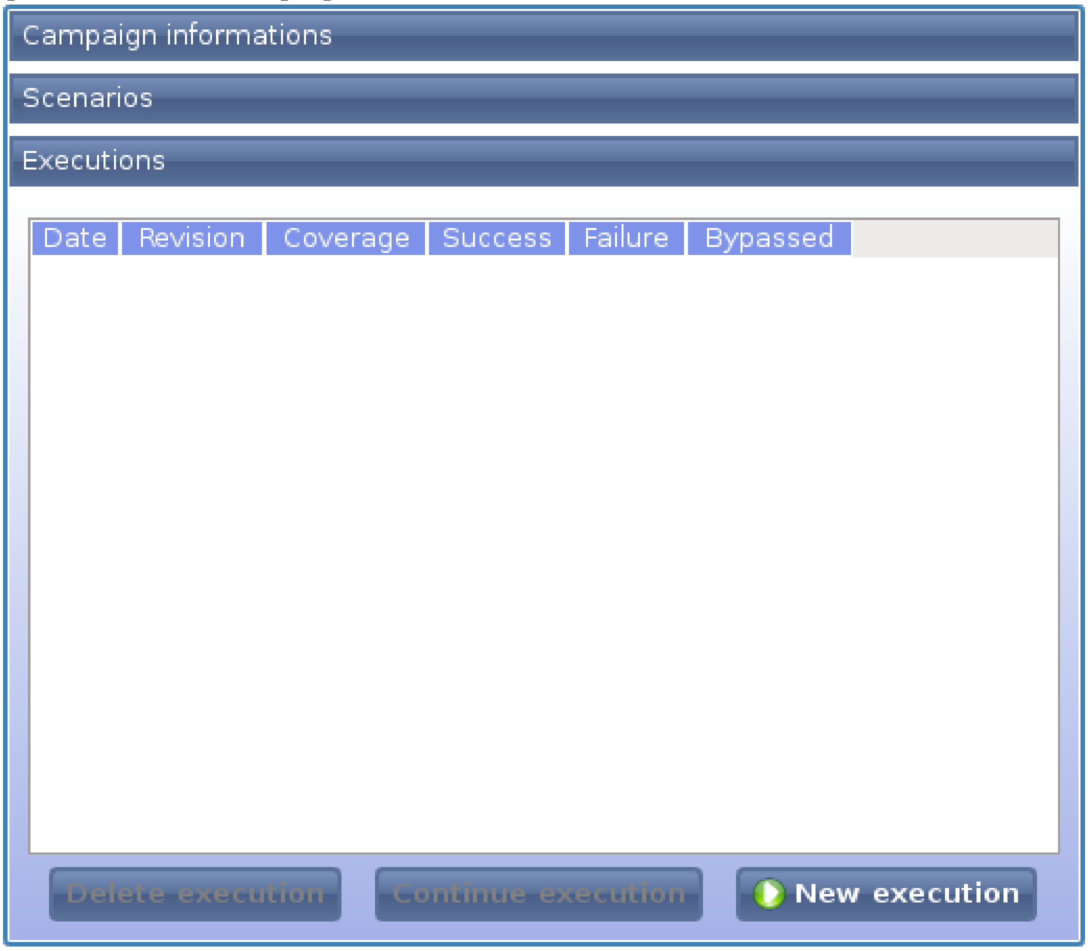
Warning deleting a campaign erases all executions associated

Manage of campaign executions

Run a campaign execution

Prerequisites : Have opened a project being connected with a user having access rights for reading on campaigns and for writing on executions campaigns (see Permissions management).

Select a campaign from the list in lower left panel of the main window of the project.
In the input screen of the campaign, select the **Executions** area.

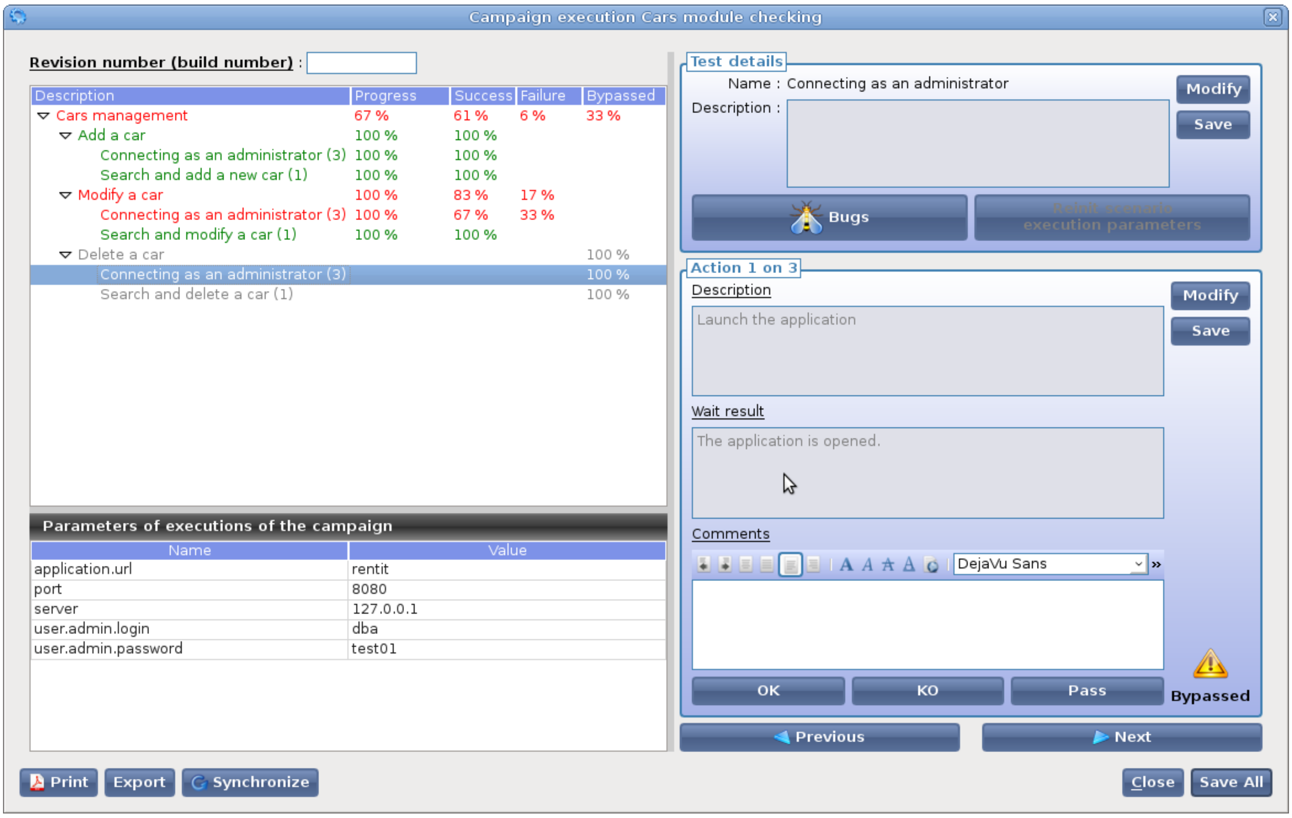


Click on [**New execution**].

The Window of a campaign execution is displayed.

Window of a campaign execution

This window is accessed when starting (see Run a campaign execution) or resuming (see Resume a campaign execution) the execution of a tests campaign.
It allows you to enter the results of the actions associated with the selected tests in the campaign. It has the following form :



The execution window is composed of :

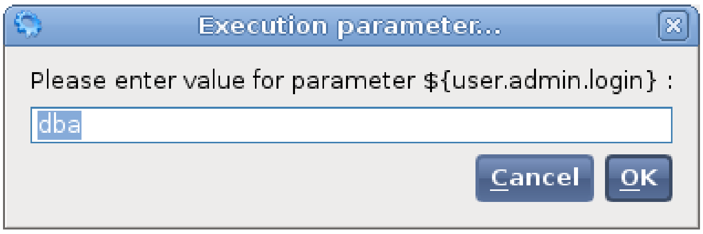
- on th top left:
  - revision number of the software tested: if this number exists, you may enter it
  - the tree of tests composing the campaign. For each test case the number of actions to be made is shown in brackets. Colors, configurable (see Colors) are associated with the status of scenarios and test cases :
    - Black: test not passed
    - blue: incomplete test
    - green: test passed
    - orange: test passed without success
- on the bottom left : the list of parameters to run the campaign: These parameters are automatically filled from the settings at the project level (see Manage project parameters).
- on the right:
  - detailed information of the running test
  - detailed information of running action: the label "**Action** 1 **of** 3" indicates that the current action is the first action of the current test that contains three actions in total.

The [**Edit**] and [**Save**] buttons in the area **Detailed information about the test** can change the test description. Executions parameters can be used in when entering new data.
The [**Edit**] and [**Save**] buttons in the area **Action X on N** can change the description and expected result of the current action. Execution parameters can be used in these fields.
The comments area allows adding comments during the execution of the action.
The [**OK**], [**KO**] and [**Change**] buttons will respectively :

- validate the result from the outcome of the current action
- Invalidate the result
- Skip action

The [**Previous**] and [**Next**] buttons permit passing to the previous and to the next action without changing comments and status (valid, invalid or past) of the action.

The [**Reset Settings execution scenario**] button permits the reset of all execution parameters used in the current scenario.
When a parameter is present in the description of a scenario or description of an action or the outcome of an action, the value of this parameter will be ask when running a scenario. A window like the following will be displayed :



Each parameter is initialized with the default value of the corresponding parameter in the campaign execution.
The scope of parameters execution of a scenario starts from the *root scenario* to which it belongs. A *root scenario* is a scenario without parent.

So, execution parameters are identical for root scenario and all its sub-scenarios.

Execution parameters may be different from a root scenario to another root scenario.

The [**Bugs**] button permits the manage of anomalies of the current test.

The [**Print**] button permits printing a detailed report on the execution of the campaign.

The [**Sync**] button permits to synchronize current running tests with tests that was possibly updated at the project or campaign, and synchronize the execution parameters with the parameters of the project.

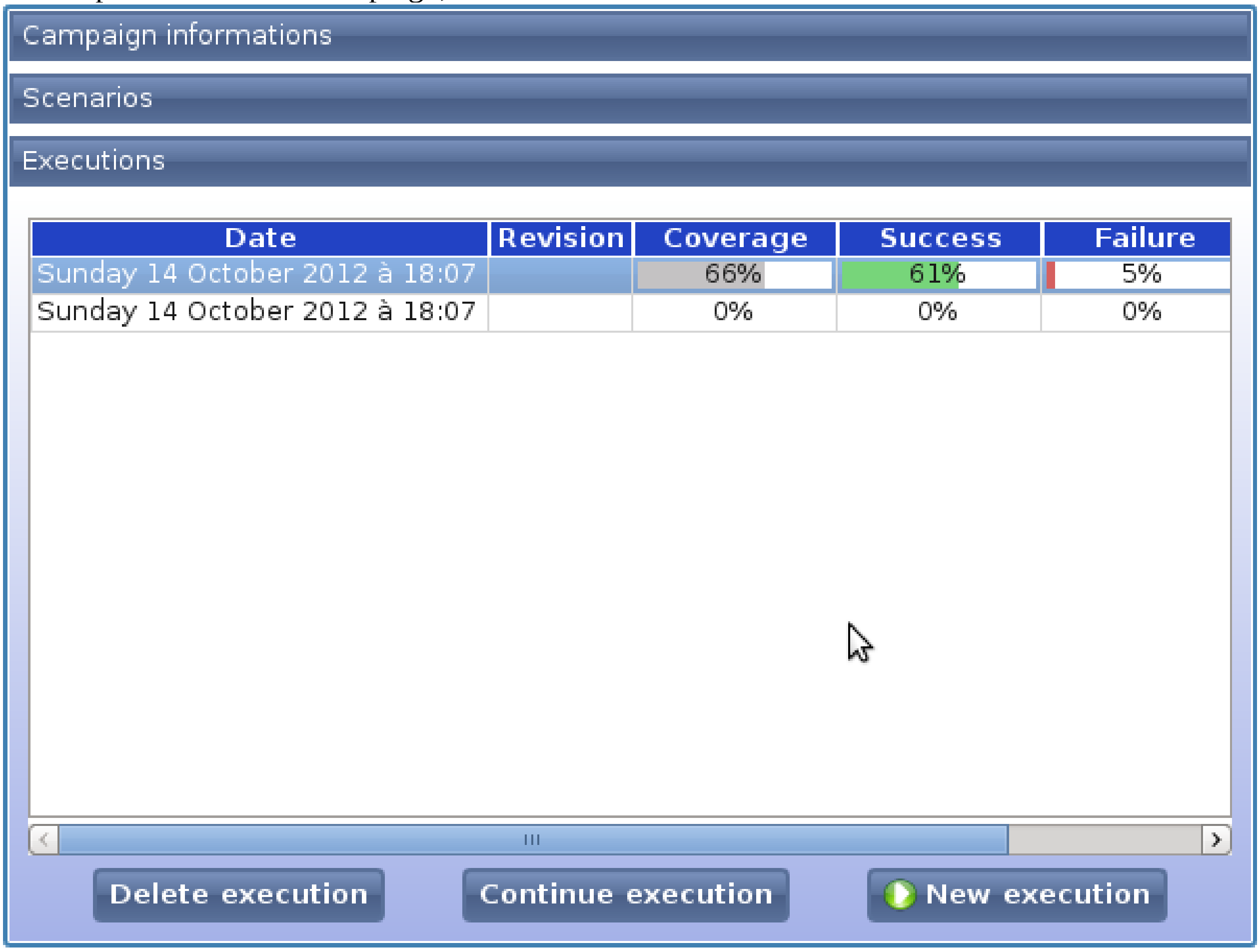
The [**Close**] button permits the close the campaign execution without saving changes.

The [**Save**] button permits to save execution modifications.

Resume a campaign execution

Prerequisites : Have opened a project being connected with a user having access rights for reading on tests campaign and for writing on executions campaigns (see Permissions management).

Select a campaign from the list in lower left panel of the main window of the project.
In the input screen of the campaign, select the **Executions** area.



Double-click on an item in the list to resume or to consult the execution of a tests campaign.

Delete a campaign execution

Prerequisites : Have opened a project being connected with a user having access rights for reading on tests campaign and for writing on executions campaigns (see Permissions management).

Select a campaign from the list in lower left panel of the main window of the project.
In the input screen of the campaign, select the **Executions** area.

Select an item in the list and click on the [**Delete execution**] button.

Add an execution campaign parameter

Prerequisites : Have launch an execution of a campaign (see Window of a campaign execution).

Click with the right mouse button in the lower left panel of the execution window and then select [**New parameter**] from the contextual menu.

A blank line appears in the parameter list:

Parameters of executions of the campaign

Name	Value
application.url	rentit
port	8080
server	127.0.0.1
user.admin.login	dba
user.admin.password	test01

Double-click on the areas **Name** and **Value** to enter data and so define the name and the value of the new execution parameter.

The parameter is automatically saved as soon as you leave the focus from input field **Name** or **Value**.

Modify an execution campaign parameter

Prerequisites : Have launch an execution of a campaign (see Window of a campaign execution).

Double-click on the areas **Name** and **Value** on a row of the lower left panel of the execution window to modify the name and the value of each parameter.

The parameter is automatically saved as soon as you leave the focus from input field **Name** or **Value**.

Delete an execution campaign parameter

Prerequisites : Have launch an execution of a campaign (see Window of a campaign execution).

Select the row of the parameter to remove on the bottom left panel of the execution window and click with the right button of the mouse and select [**Delete selected parameter**] from the contextual menu.

Caution: Deleting a parameter is immediate.

Add a bug when executing a test

Prerequisites : Have launch an execution of a campaign (see Window of a campaign execution).

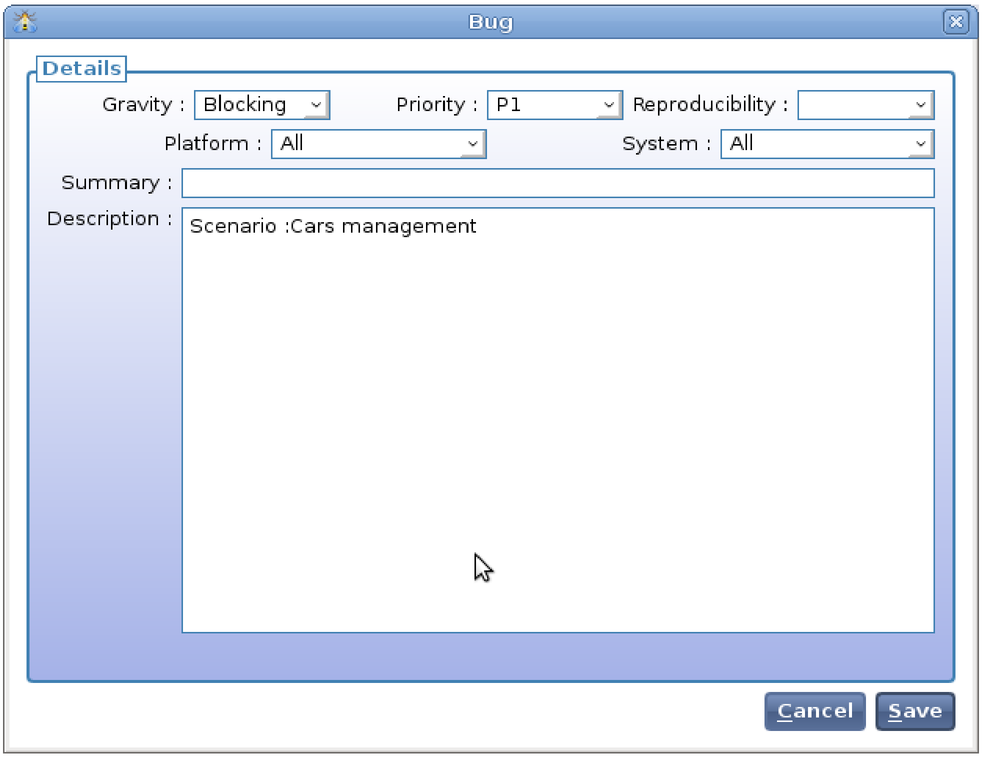
Area on a detailed scenario / test case has a button [**Bugs**].:

Click this button to display the list of bugs recorded in RTMR for selected scenario / test case :

Click on [**Add a bug**] to display the main screen of a bug .

Entering an bug in the internal bugtracker

If the version of the current project is **not** associated with an external bugtracker (see Modify project properties), the following screen appears :

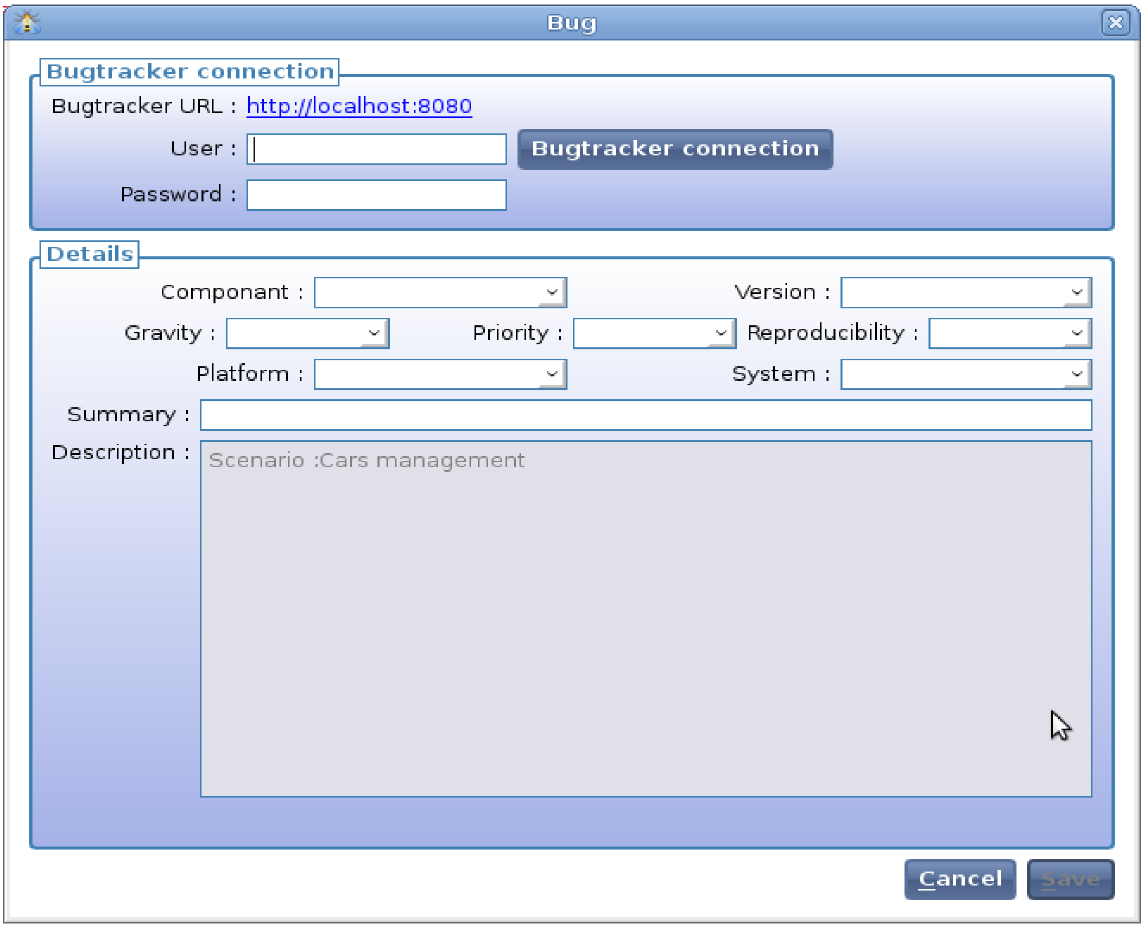


Enter the required values for registration of the bug :

- Severity : list of types of gravity
- Priority : list of the types of priorities
- Repeatability : list of types of reproducibilities
- Platform : list of types of platforms
- System : list of types of systems
- Summary : This is a summary of the bug (eg "car registration error")
- Description : This is a detailed description of the bug (eg, "when the car model name exceeds 100 characters the error 10000 is displayed")

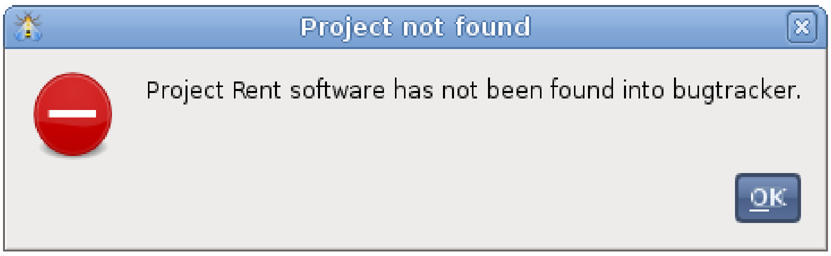
Entering an bug in the external bugtracker

If the version of the current project is associated with an external bugtracker like Bugzilla (see Configuration of Bugzilla) or like Mantis (see Configuration of Mantis), the following screen appears :



The **Bugtracker connection** area allows users to authenticate with the bugtracker.
To avoid having to retype **User** and **Password** for each bug, those informations can be preset (see Bugtrackers).

Once those informations are entered, click on [**Log bugtracker**].
This has the effect of load dynamically (via the webservice of the bugtracker) drop-down menus containing the informations necessary to register a bug.
If the project is not found in the bugtracker, recording the bug can not be performed and the following error message appears:



In this case check the bugtracker parameters defined at the project level (see Configuration of Bugzilla or Configuration of Mantis).

If the current version of the project is not found in the bugtracker, the following warning message appears :

This message is not blocking and bug may be registered anyway.
However, for a better monitoring of bugs, it is recommended that you create for each version of your project a corresponding version in the targetted bugtracker.

Loaded down menus, enter the values required for registration of the bug :

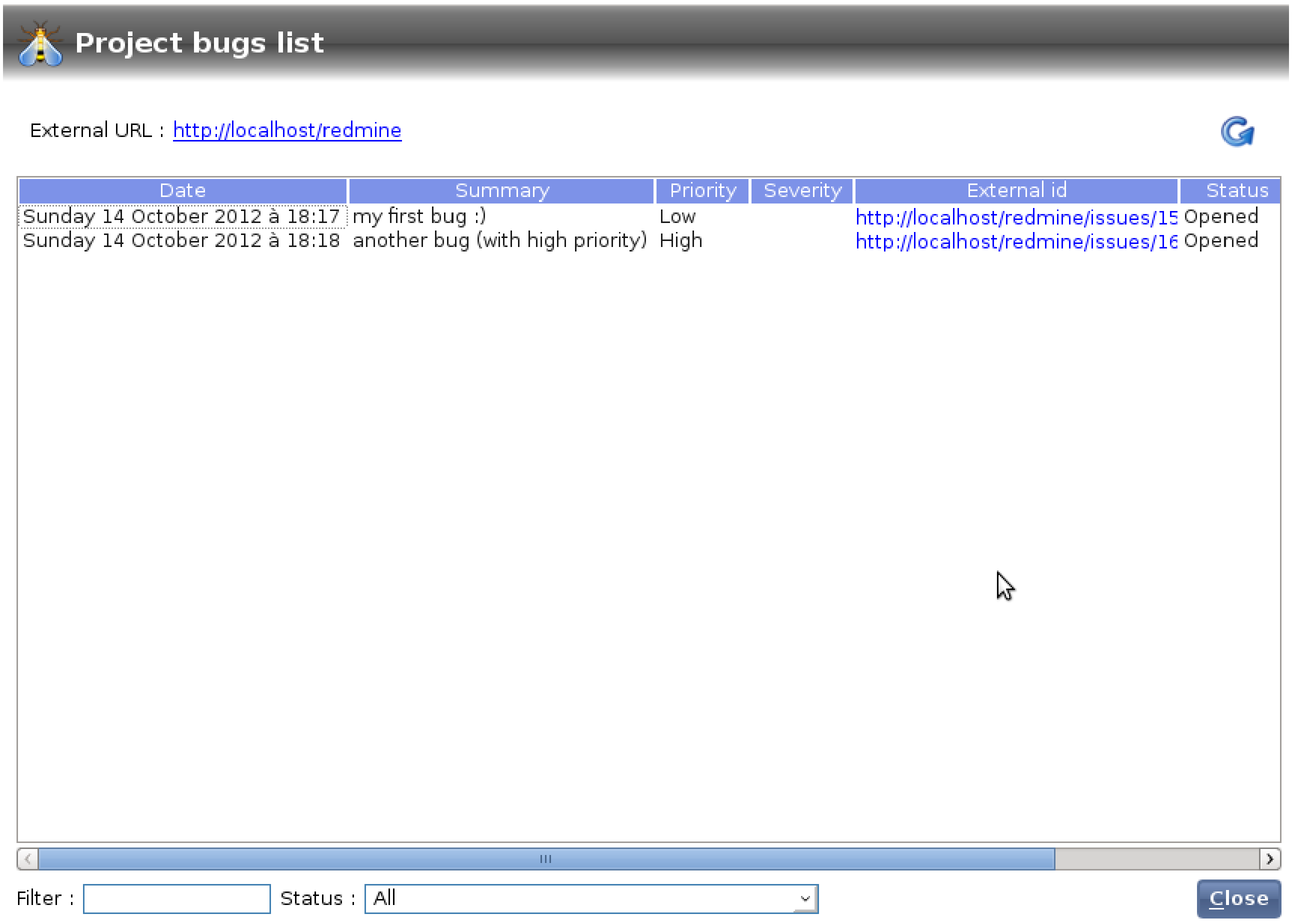
- Component : these are the components of the project defined in the bugtracker (eg: GUI, server, database etc ...)
- Version: these are the versions of the project defined in the bugtracker
- Severity : list of types of gravity of the bugtracker
- Priority : list of the types of priorities of the bugtracker
- Repeatability : list of types of reproducibilities of the bugtracker
- Platform : list of types of platforms of the bugtracker
- System : list of types of systems of the bugtracker
- Summary : This is a summary of the bug (eg "car registration error")
- Description : This is a detailed description of the bug (eg, "when the car model name exceeds 100 characters the error 10000 is displayed")

Manage project bugs

Prerequisites : Have opened a project being connected with a user having access rights on project (see Permissions management).

Select the menu [**Project**][**Bugs**].

The following screen appears :



If the version of the opened project is associated with an external bugtracker, the webservice URL of the bugtracker is displayed at the top of the window.
The button on the right allows you to synchronize local bugs data with remote bugs data recorded in the bugtracker. The synchronisatino is using the authentication information provided from the Bugtrackers options.

The Filter text box can display only anomalies whose name or summary contains the entered text.

The status list may filter opened bugs, closed bugs or any other bugs.

Manage project parameters

The project parameters are used first when entering test cases (see Manage actions of a test case). They are useful to refer to variable information such as :

- data related to a technical environment (web address, server address, port number, user name, password ... etc)
- data related to a functional environment: in this case we speak of the dataset (eg parameters related to a software that manages of cars rental : car registration, account number, insurance policy...etc)

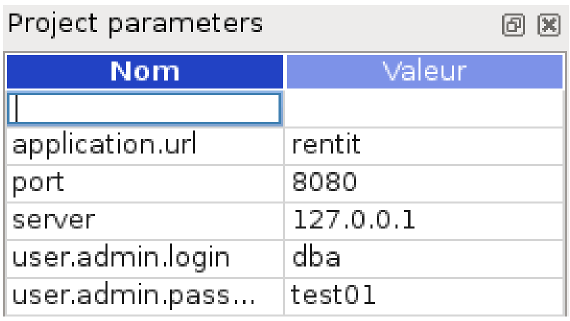
These parameters are then evaluated during the execution of the tests campaign (see Window of a campaign execution).

Add a project parameter

Prerequisites : Have opened a project being connected with a user having access rights to the project and a user having a writer role (see Permissions management).

Select the menu [**Project**][**New parameter**] or click with the right mouse button in the lower right side panel and select [**New parameter**] from the contextual menu.

A blank line appears in the parameter list :



Double-click on the areas **Name** and **Value** to enter data and so define the name and the value of the new execution parameter.

The parameter is automatically saved as soon as you leave the focus from input field **Name** or **Value**.

Modify a project parameter

Prerequisites : Have opened a project being connected with a user having access rights to the project and a user having a writer role (see Permissions management).

Double-click on the areas **Name** and **Value** on a row of the lower left panel of the execution window to modify the name and the value of each parameter.

The parameter is automatically saved as soon as you leave the focus from input field **Name** or **Value**.

Delete a project parameter

Prerequisites : Have opened a project being connected with a user having access rights to the project and a user having a writer role (see Permissions management).

Select the row of the parameter to remove on the bottom left panel of the main window and select the menu [**Project**][**Delete the selected parameter**]or click with the right button of the mouse on the row of the parameter to remove and select [**Delete selected parameter**] from the contextual menu.

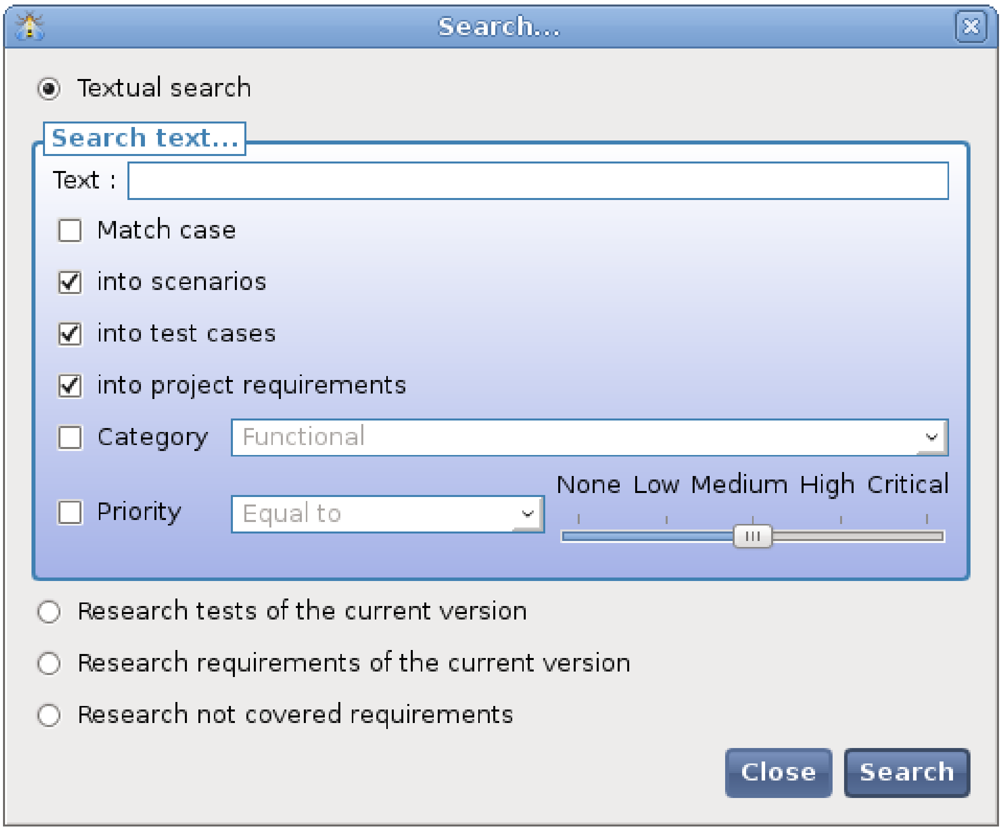
Caution: Deleting a parameter is immediate.

Search

A search form allows you to search requirements and scenarios in the project.

Prerequisites : Have opened a project.

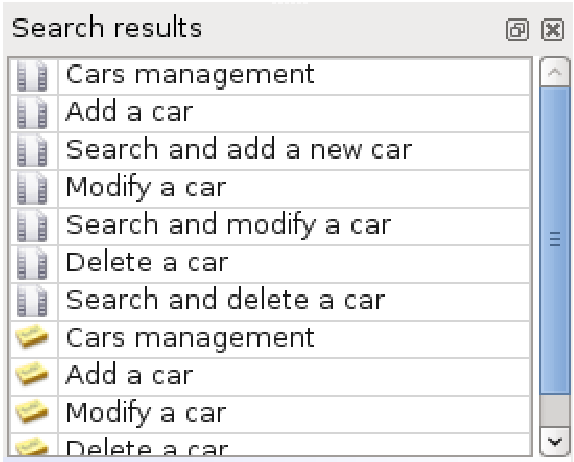
Select the menu [**Project**][**Search**]. The search form is displayed :



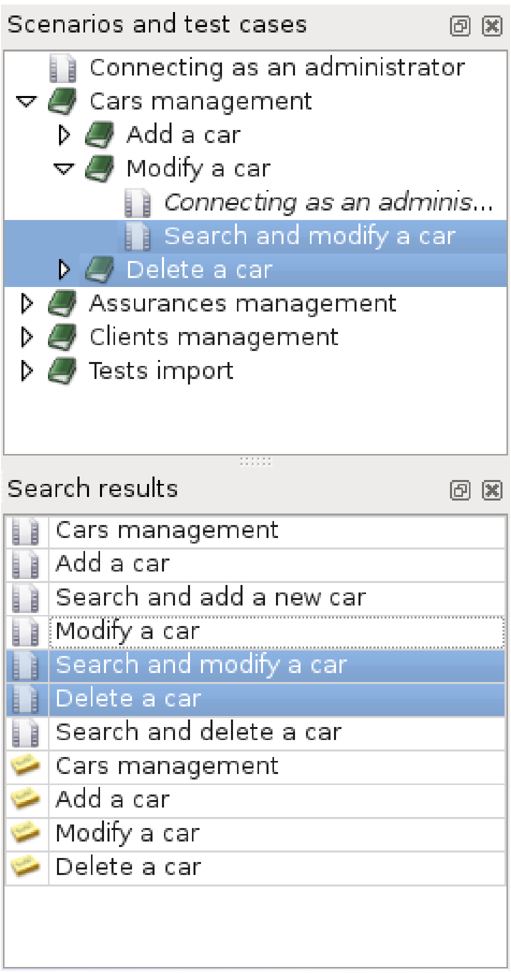
Use this form to search :

- requirements, scenarios and test cases including
  - the content (name, description and wait result for the actions of test cases) refers to the text entered in the text box to search
  - the category corresponds to the selected category
  - the priority is the selected priority
- tests of the current version
- requirements of the current version
- requirements not covered: ie the requirements that are associated with any tests

Make a choice then click on [**Search**].
The results of the search are displayed in a panel that appears at the bottom right of the main window of the project. Example search result for the word “*car*”:



In this example, the word *car* was found in four requirements and several scenarios.
To easily locate the requirements and scenarios found in their respective tree, select the rows in the results pane using the mouse and the [**Ctrl**] and [**Maj**] key. This has the effect of automatically selecting the associated requirements and scenarios.
Example of lines selection "*Search and add a car*" and "*Edit a car* ' :



This type of manipulation is very useful to create a test campaign quickly :

- research on the requirements and scenarios to cover

- select results from the search panel
- drag/drop requirements and / or scenarios automatically selected from their tree to the input screen of the test campaign previously opened

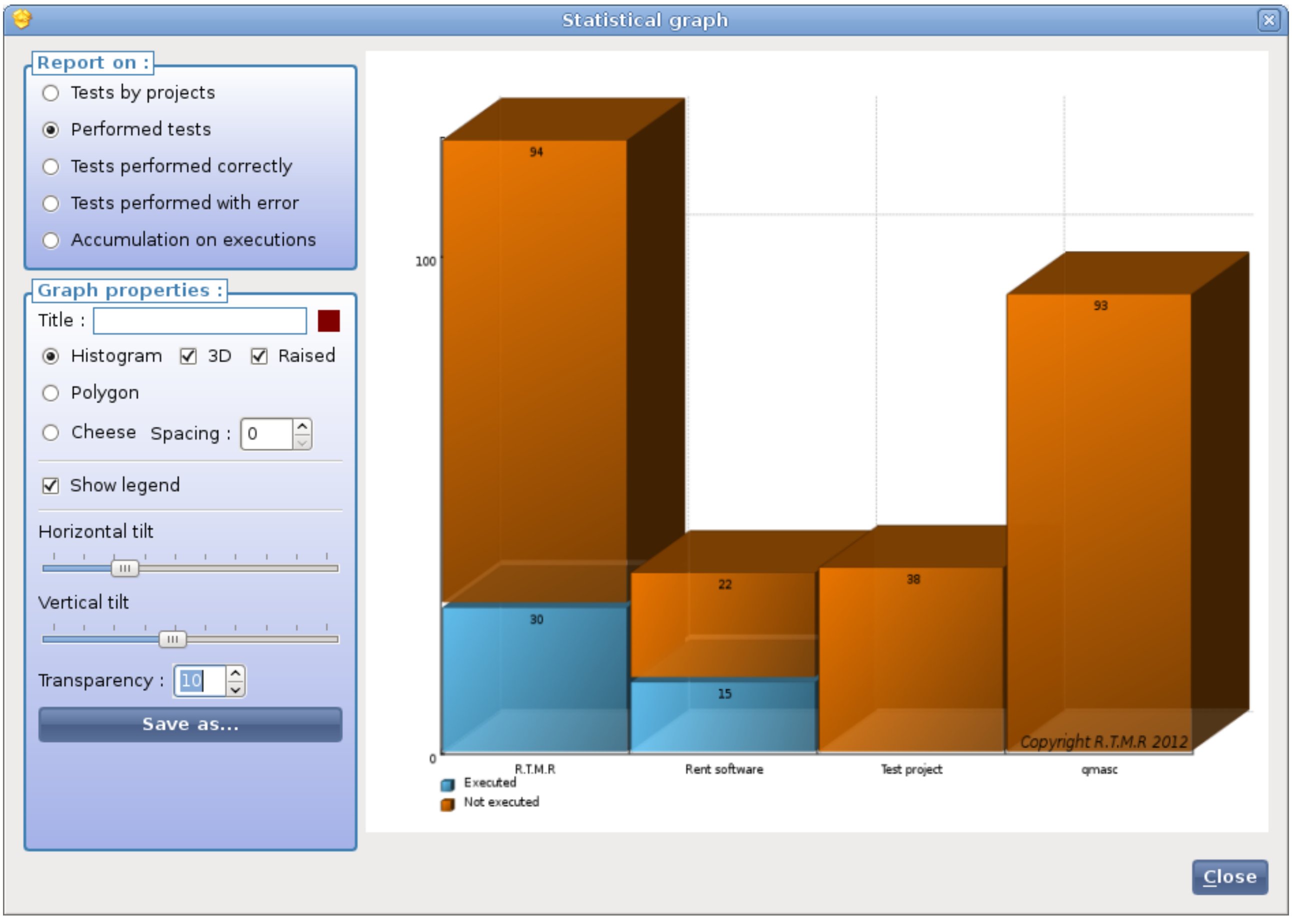
In addition, in the panel search results, when double-clicking on the line of a result, the input screen of the requirement or scenario is directly displayed.

Reporting

To view reports on a particular project refer to Executions reports of projects.

To view reports on all projects you must be connected with an administrator role (see Permissions management).
Select the menu [**Administration**][**Project reports**].

The following screen appears :



Reporting options are :

- tests by projects : displays the number of tests by projects
- performed tests : displays the number of tests present in a test campaign (by projects)
- tests performed correctly : displays the number of tests performed correctly for each project
- tests performed with error : displays the number of tests performed with an error for each project
- accumulation of executions : displays the status of tests performed for each project

Navigation

Prerequisites : Have open a project.

The [**Navigation**] achieves directly :

- the homepage of the project : it is the project properties screen
- previous page : this is the screen that was previously accessed
- next page : this is the next screen viewed in the browser history

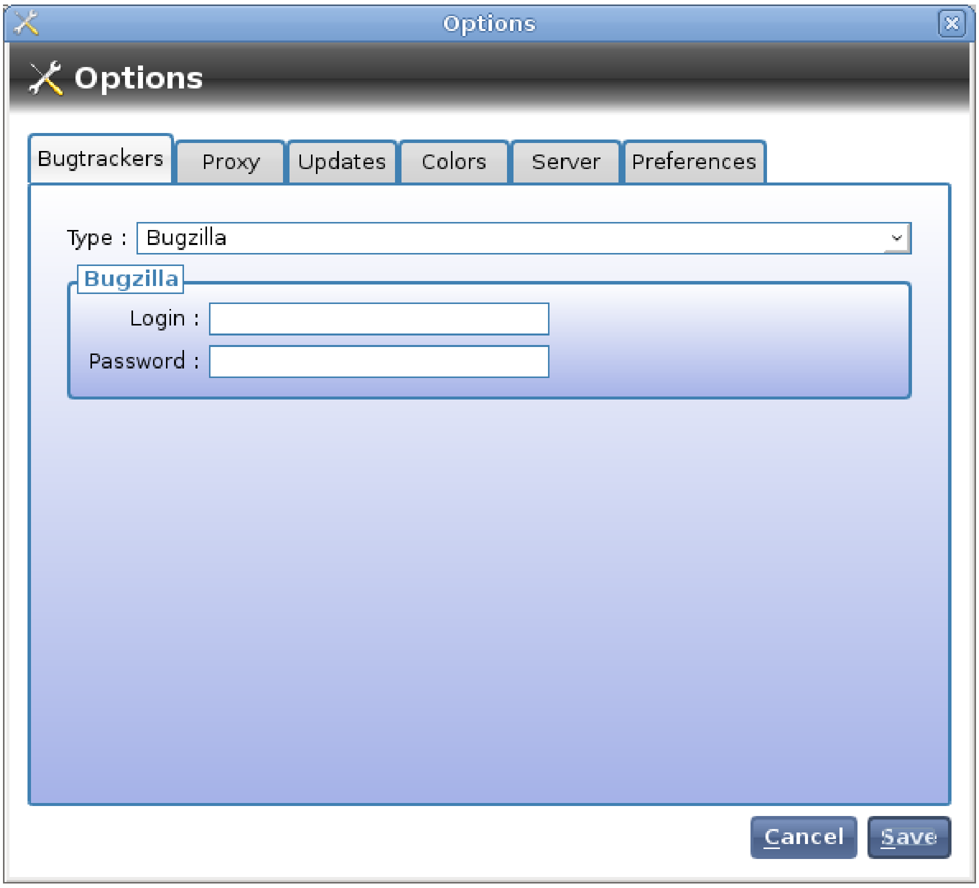
Options

Bugtrackers

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**].

The following screen appears :



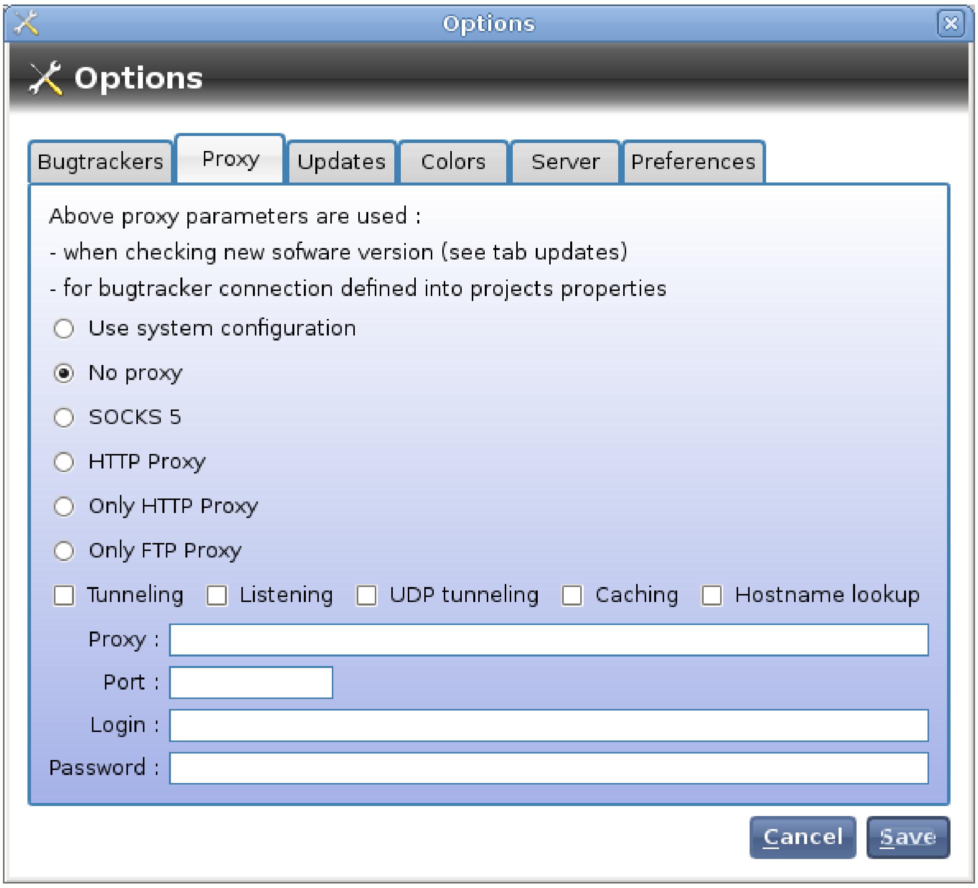
Bugtrackers tab allows you to configure authentication information for bugtrackers like Bugzilla, Mantis, Redmine...etc.
This informations will be pre-selected when entering an anomaly and used for synchronization between anomalies of the open project and the bugtracker anomalies (see Manage project bugs).

Proxy

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**], then selecy **Proxy** tab.

The following screen appears :



This informations are used for :

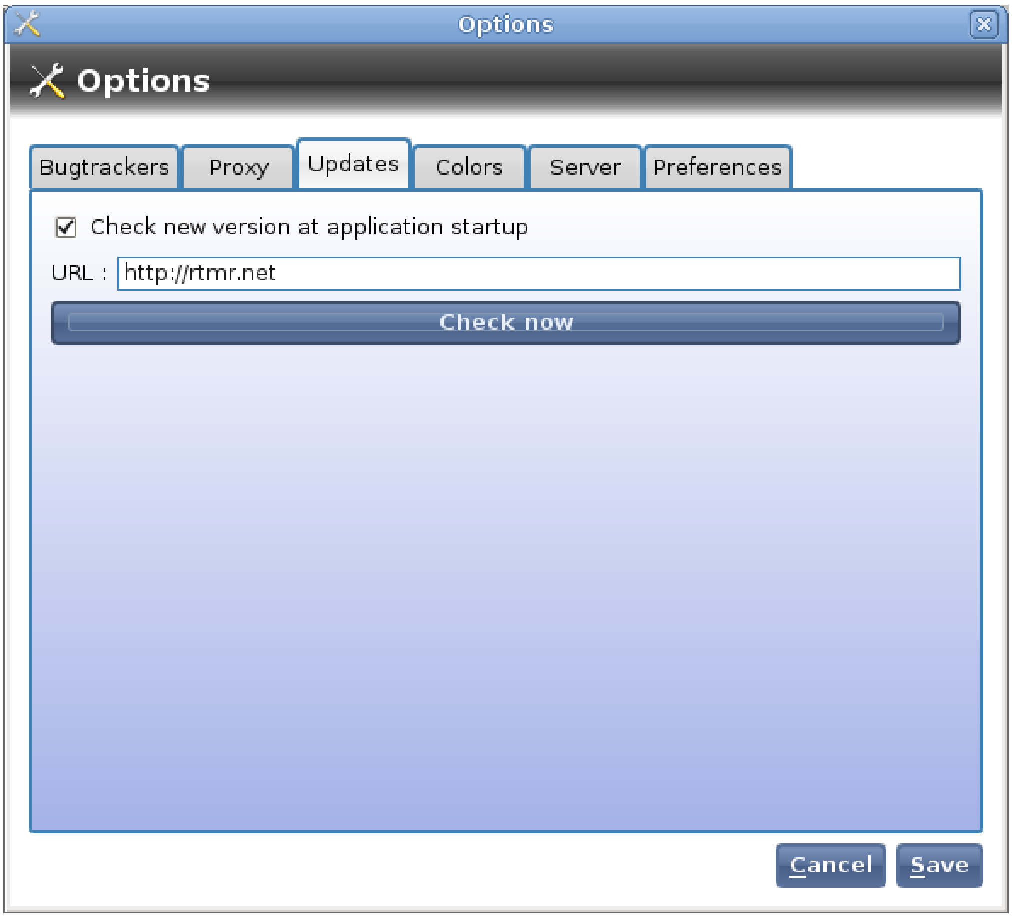
- verification of new software version (see Update)
- connect to external bugtracker defined in the project properties

Update

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**], then select **Update** tab.

The following screen appears :



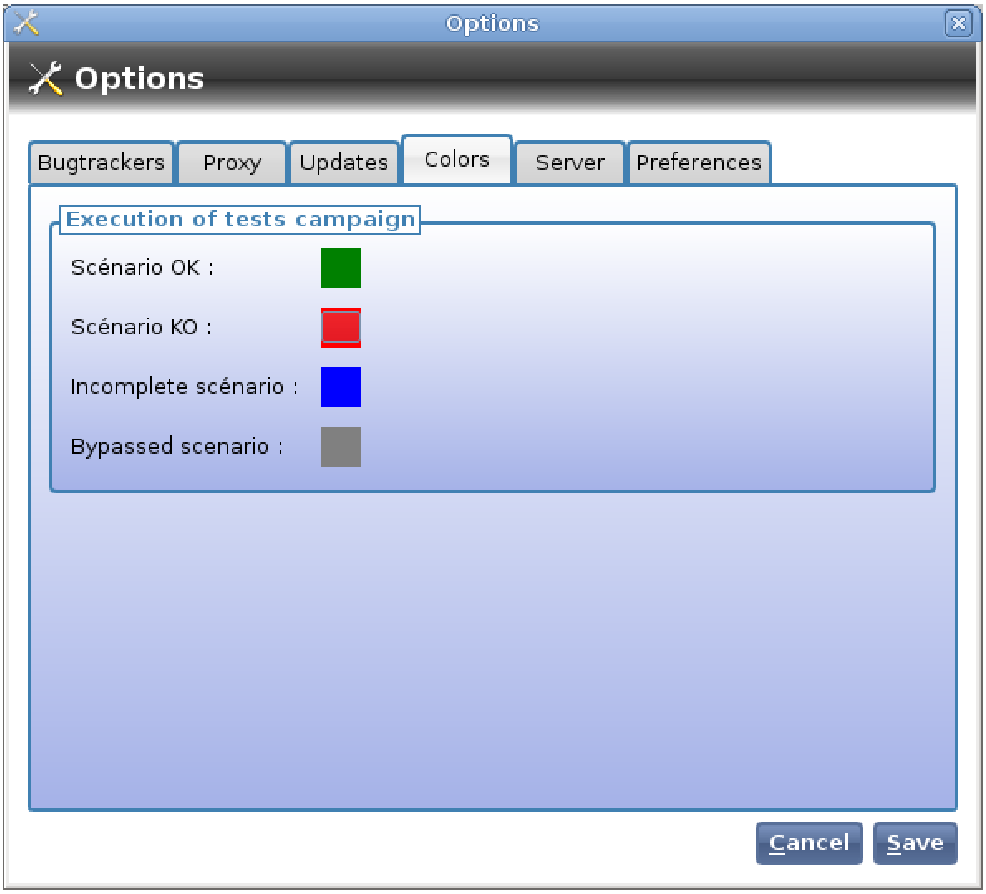
This screen allows you to automatically check the availability of a new version of the client on site RTMR defined at the **URL**

Colors

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**], then select **Colors** tab.

The following screen appears :



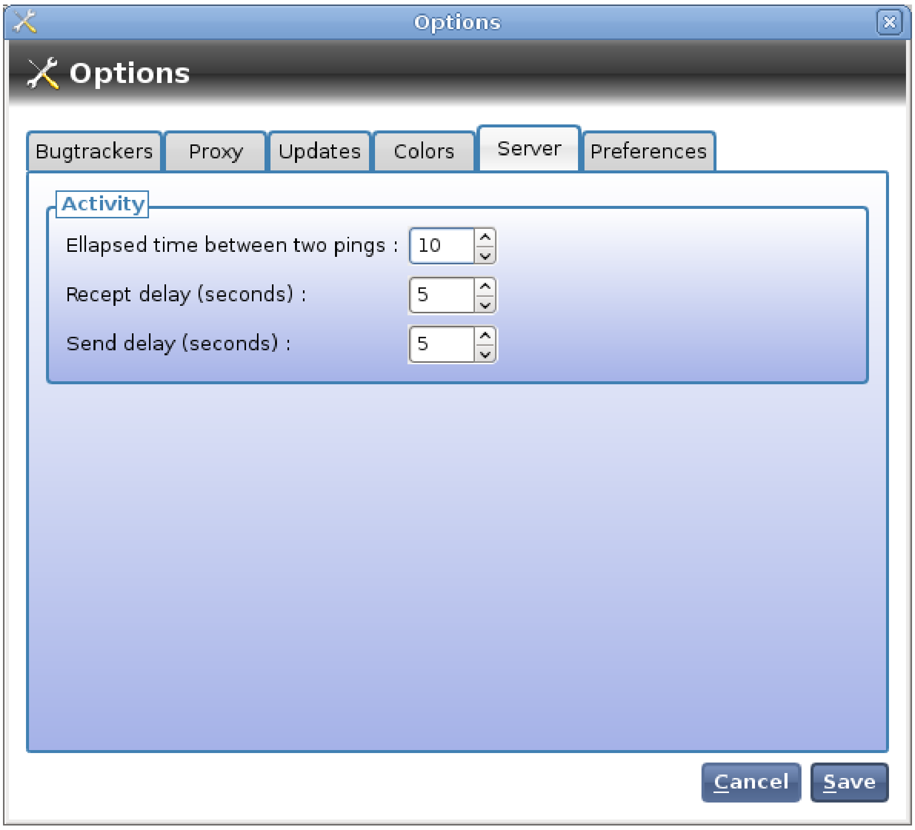
Click on the colored squares to customize colors that are used in graphics and during the execution of a test campaign (see Window of a campaign execution).

Server

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**], then select **Server** tab.

The following screen appears :



This tab may be use to customize netserver connexion delays.

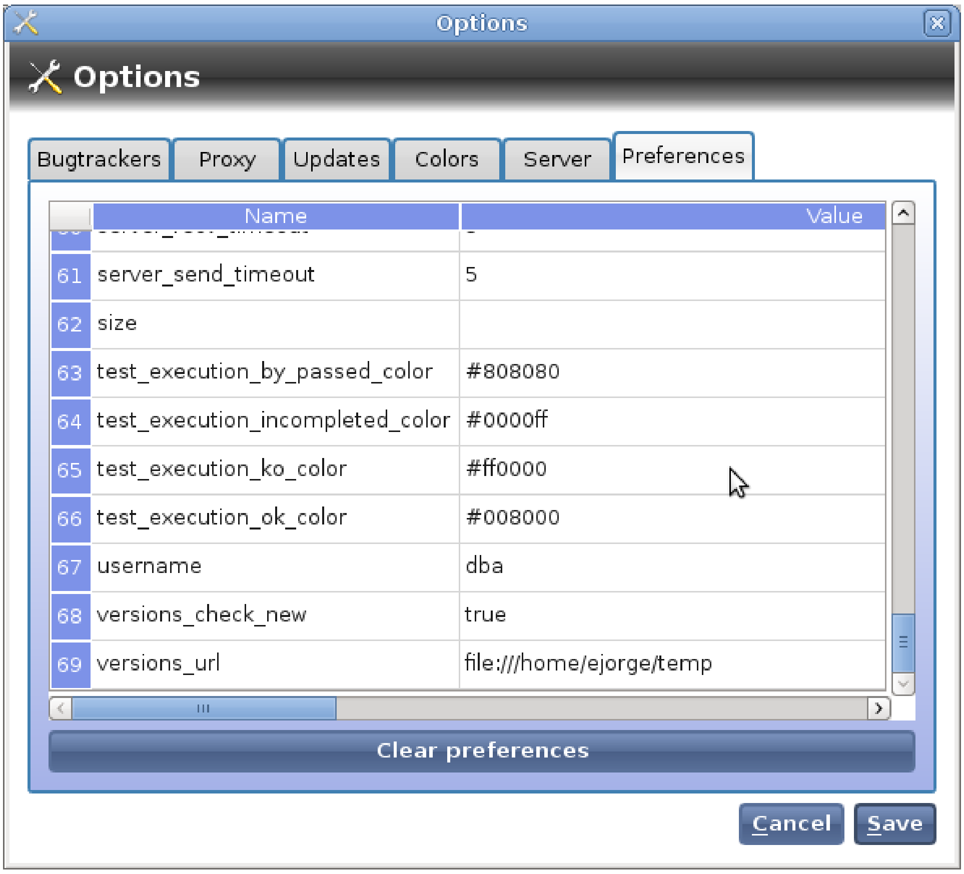
Increase *Recept delay* and *Send delay* if you network has a low bandwidth.

Preferences

Prerequisites : Be connected (see Permissions management).

Select the menu [**Tools**][**Options**], then select **Preferences** tab.

The following screen appears :



This screen shows all preferences uses by rtmr application, like main window position, colors, bugtracker informations...etc.

The button [**Clear preferences**] may be use to erase all preferences.

External modules (plug-ins)

Since version 1.9.0 of the RTMR client the possibility to extend the functionality of the application is possible through the development of plug-ins.
This change allows create plugins of various types:

- bugtracker plugin (to manage anomalies)
- projects plugin
- requirements plugin
- tests plugin
- campaigns plugin

Version 1.9.0 is also distributed with two bugtracker plugins that are Bugzilla and Mantis. These modules were integrated directly in the application and have been outsourced.

The plugins are libraries that are loaded at application launch from the *modules* directory of the client