Integrating SVN checkout and update facilities in the CAPRI GUI to ease installation and maintenance of CAPRI

- Wolfgang Britz, July 2010 -

Introduction

CAPRI is hosted on the SVN software versioning system (see e.g. http://en.wikipedia.org/wiki/Apache_Subversion) which ensures that CAPRI users and developers can operate smoothly in a distributed network. For developers who need to upload changes made to CAPRI code to the server (a process called “commit”), TortoiseSVN (http://tortoisesvn.tigris.org/) is the recommended tool. TortoiseSVN is integrated nicely into windows, but it might take a while until the logic behind the SVN operations is fully understood by a novice user.

For users which do not contribute to the code basis of CAPRI or use TortoiseSVN in other contexts, installing and learning to master TortoiseSVN as an additional tool is an unnecessary burden. Therefore, the client based SVN basic operations which allow a user to keep its local copy synchronized with the server are now embedded in the java code of the GUI. For those who only need read-only access to the CAPRI server repository, an installation of TortoiseSVN is not longer necessary.

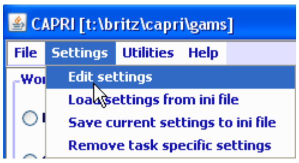
The changes necessary in the GUI can be summarized as follows. Firstly, new SVN related entries in the initialisation file can be edited by the user. And secondly, a new dialogue allows starting an update. The following sections give a quick overview over the new functionalities.

Case one: Exploiter and runner

Entering the necessary information to link to the SVN server

An exploiter by definition only accesses GDX files from the result directory. He is not allowed to run GAMS programs, and thus do not need access to the GAMS source code, data and restart files read in by the different GAMS based working steps of CAPRI.

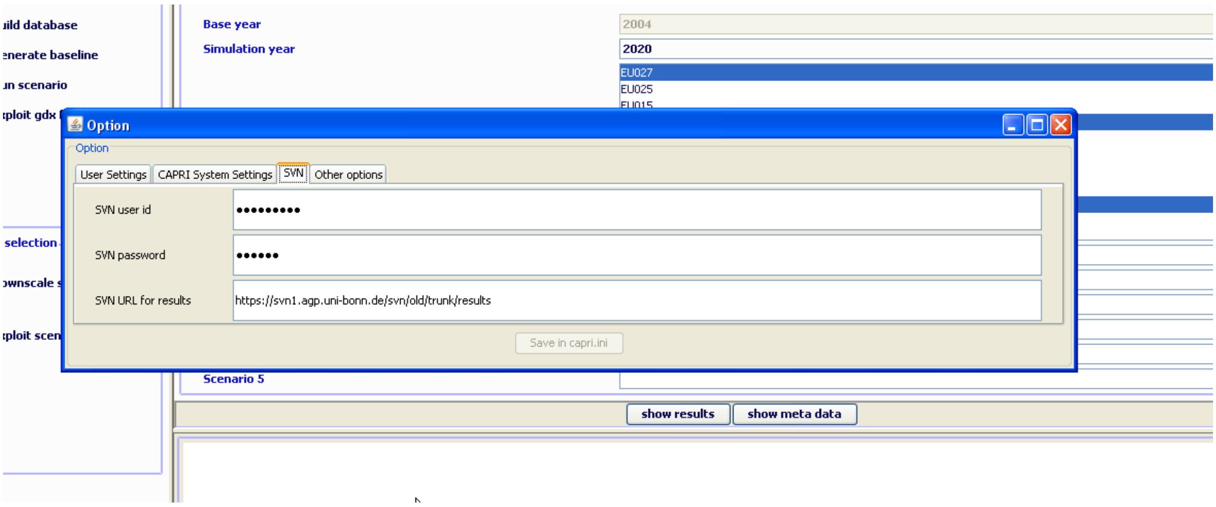
Accordingly, in order to work with SVN, only three pieces of information have to be entered under “Settings / Edit Settings” in the SVN tab:


CAPRI [t:\britz\capri\gams]
File
Settings
Utilities
Help
Edit settings
Load settings from ini file
Save current settings to ini file
Remove task specific settings

- The SVN user id
- The SVN password
- The url of the result directory

The first two fields are not visible, and the related entries in the ini file are encrypted. The last entry can be set to a specific branch relating e.g. to a training session. That allows for CAPRI “mini installations”. These mini installations do not need to be distributed as SVN installations as the SVN interface in the GUI will also allow to checkout over existing sub-directories and files. That ensures some additional safety regarding access information to sensible branches of the server – a bystander cannot read the user id and password. But users should always place local copies of such branches including the directory from which the GUI is started on secured parts of their file system.

The local directory for the GUI is simply taken from the start directory of the GUI; whereas the SVN address for the GUI is stored in the “default.ini” file.

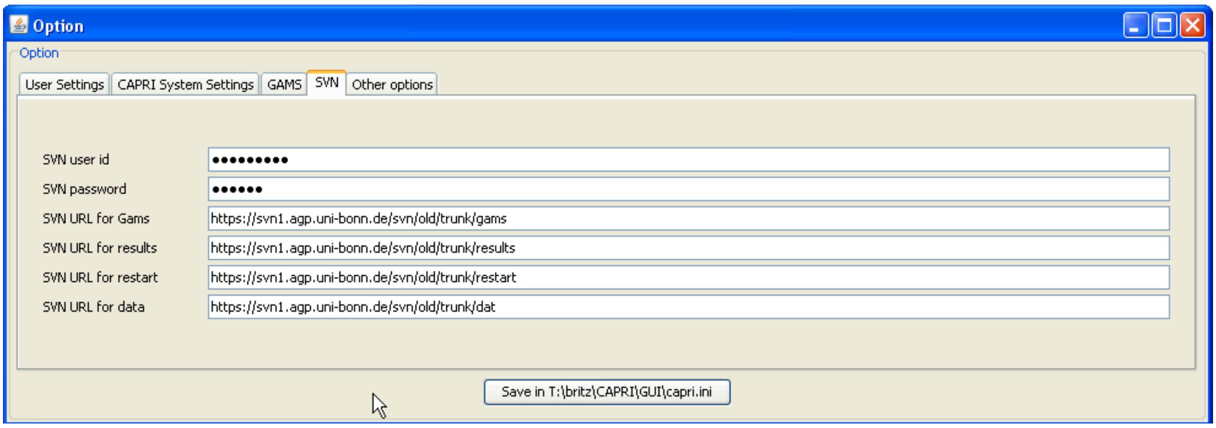


SVN user id ●●●●●●●●●

SVN password ●●●●●●

SVN URL for results https://svn1.agp.uni-bonn.de/svn/old/trunk/results

The runner can enter the additional SVN urls relating to the different sub-directories of a CAPRI installation. That should give some flexibility when working with branches on the server:



SVN user id: ●●●●●●●●●

SVN password: ●●●●●●

SVN URL for Gams: https://svn1.agp.uni-bonn.de/svn/old/trunk/gams

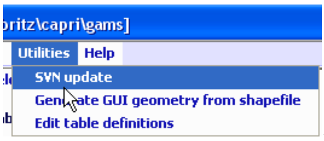
SVN URL for results: https://svn1.agp.uni-bonn.de/svn/old/trunk/results

SVN URL for restart: https://svn1.agp.uni-bonn.de/svn/old/trunk/restart

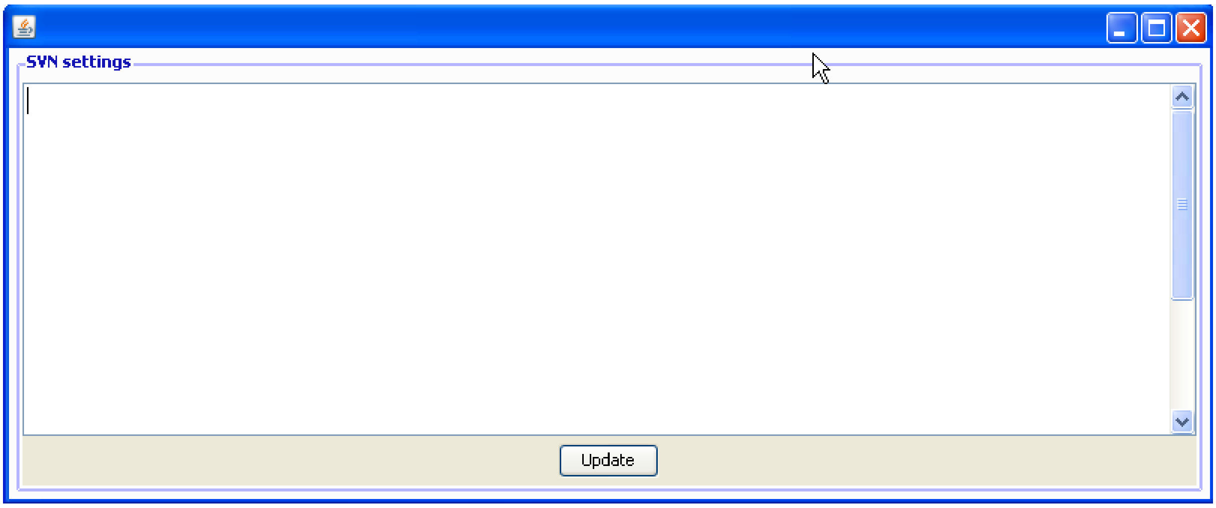
SVN URL for data: https://svn1.agp.uni-bonn.de/svn/old/trunk/dat

Performing an update

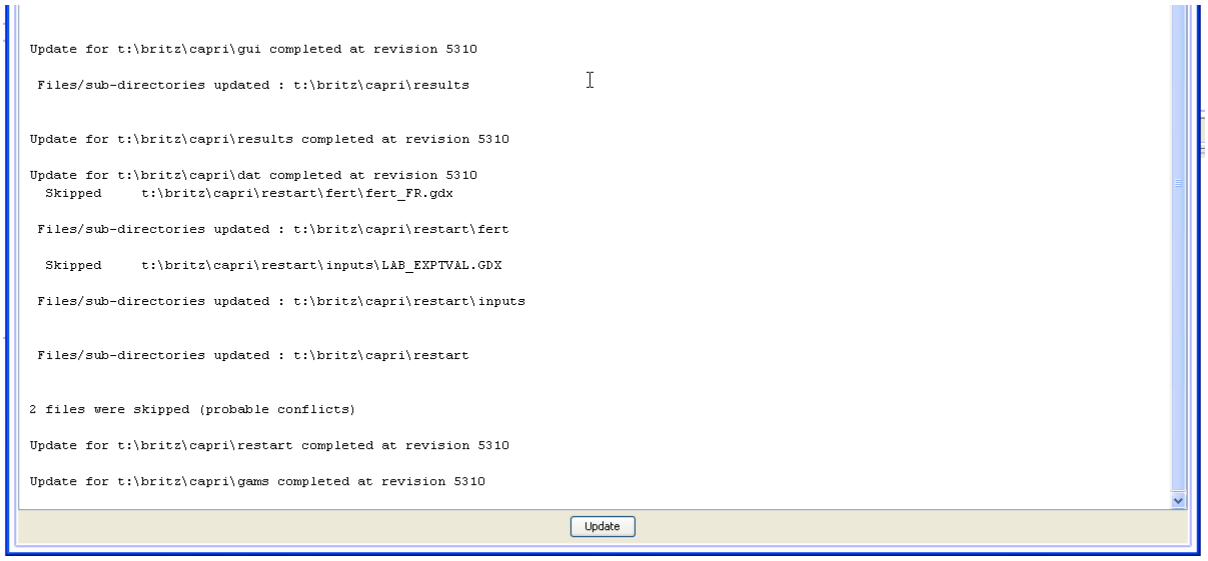
The second functionality for an exploiter (and runner) is to update all directories with the menu item "Utilities / SVN update" . An update will download updated versions of files into hidden directories, and, if the related files in the local working copy has not been modified, will also replaced the local files.



Choosing that menu item will open a dialogue with just one button termed "update" and an area into which messages from the SVN updates / checkouts are reported:



Pressing the "update" bottom will trigger an. Possible conflicts, merges etc. are shown in the reporting area:

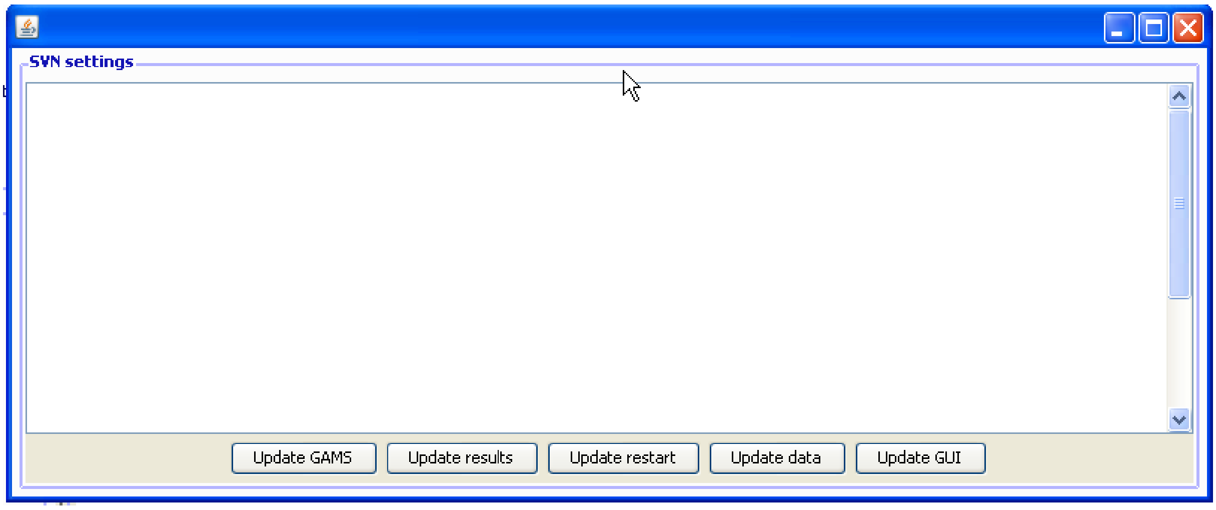


```
Update for t:\britz\capri\gui completed at revision 5310

 Files/sub-directories updated : t:\britz\capri\results


Update for t:\britz\capri\results completed at revision 5310

Update for t:\britz\capri\dat completed at revision 5310
  Skipped     t:\britz\capri\restart\fert\fert_FR.gdx

 Files/sub-directories updated : t:\britz\capri\restart\fert

  Skipped     t:\britz\capri\restart\inputs\LAB_EXPTVAL.GDX

 Files/sub-directories updated : t:\britz\capri\restart\inputs


 Files/sub-directories updated : t:\britz\capri\restart


2 files were skipped (probable conflicts)

Update for t:\britz\capri\restart completed at revision 5310

Update for t:\britz\capri\gams completed at revision 5310
```

If the directory is not yet under version control, the GUI will perform a checkout instead, i.e. setting up the first installation of the hidden copies from the server. Before an update, a “clean-up” operation will remove any possible local locks related to earlier unsuccessful SVN operations. That should ensure that in most cases smooth updates as long as an internet connection is available, and avoid some of the more tricky problems TortoiseSVN users might face.

Case two: Administrator

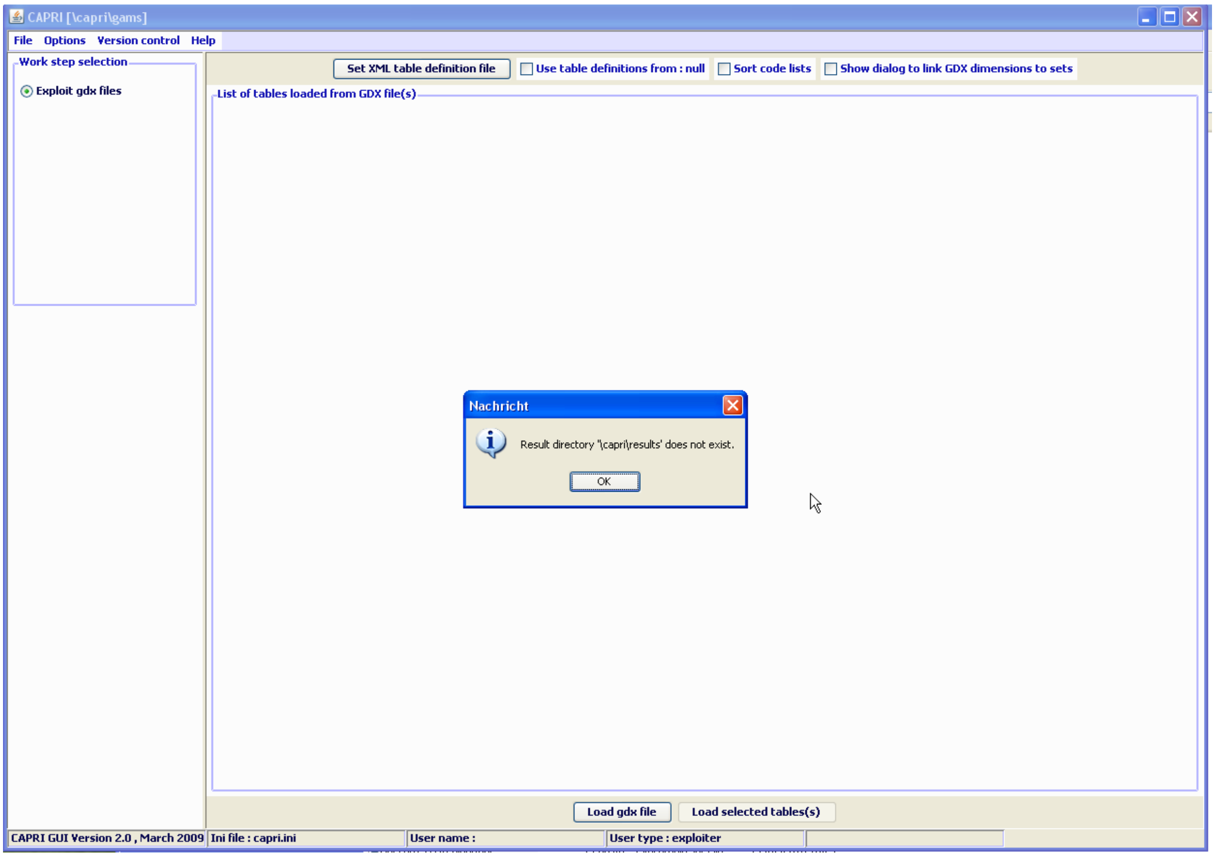
An administrator can enter the same SVN directories as a runner, but can trigger updates for the different parts separately:



Usage for installation purposes

Since quite a while the CAPRI network discusses how installations specifically for training sessions can be organized more easily. The newly embedded SVN functionalities in the GUI should ease that task somewhat, specifically in cases where only exploitation functionalities are asked for.

The installation of CAPRI based on the new functionality is relatively straightforward. As before, a JAVA run time engine must be installed for the GUI to run. For an exploiter, only a minimum GUI installation (e.g. without the large geometries for the 1x 1 km layer) and the necessary results files to view can then be copied to a local directory. At first start, the user must then only enter where the results had been copied to (if the result files are not parallel to the GUI) and save the information to his new *CAPRI.INI* file.



○ Exploit gdx files

☐ Use table definitions from : null

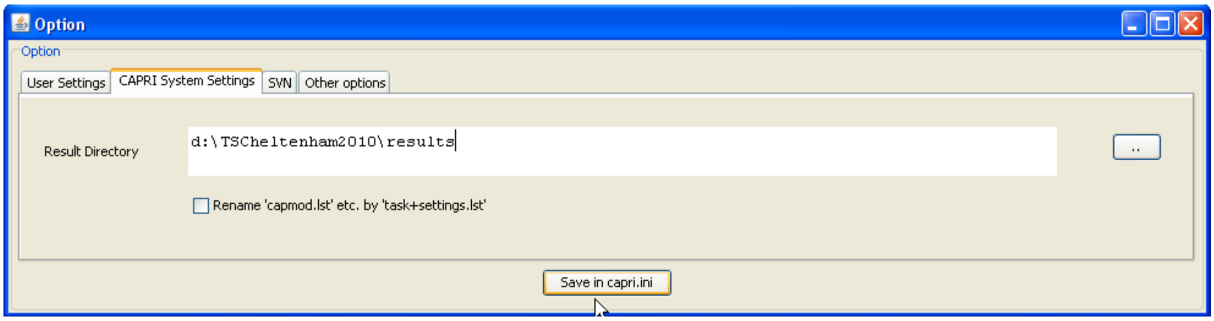
☐ Sort code lists

☐ Show dialog to link GDX dimensions to sets

Result directory "\capri\results' does not exist.

CAPRI GUI Version 2.0 , March 2009 | Ini file : capri.ini | User name : | User type : exploiter

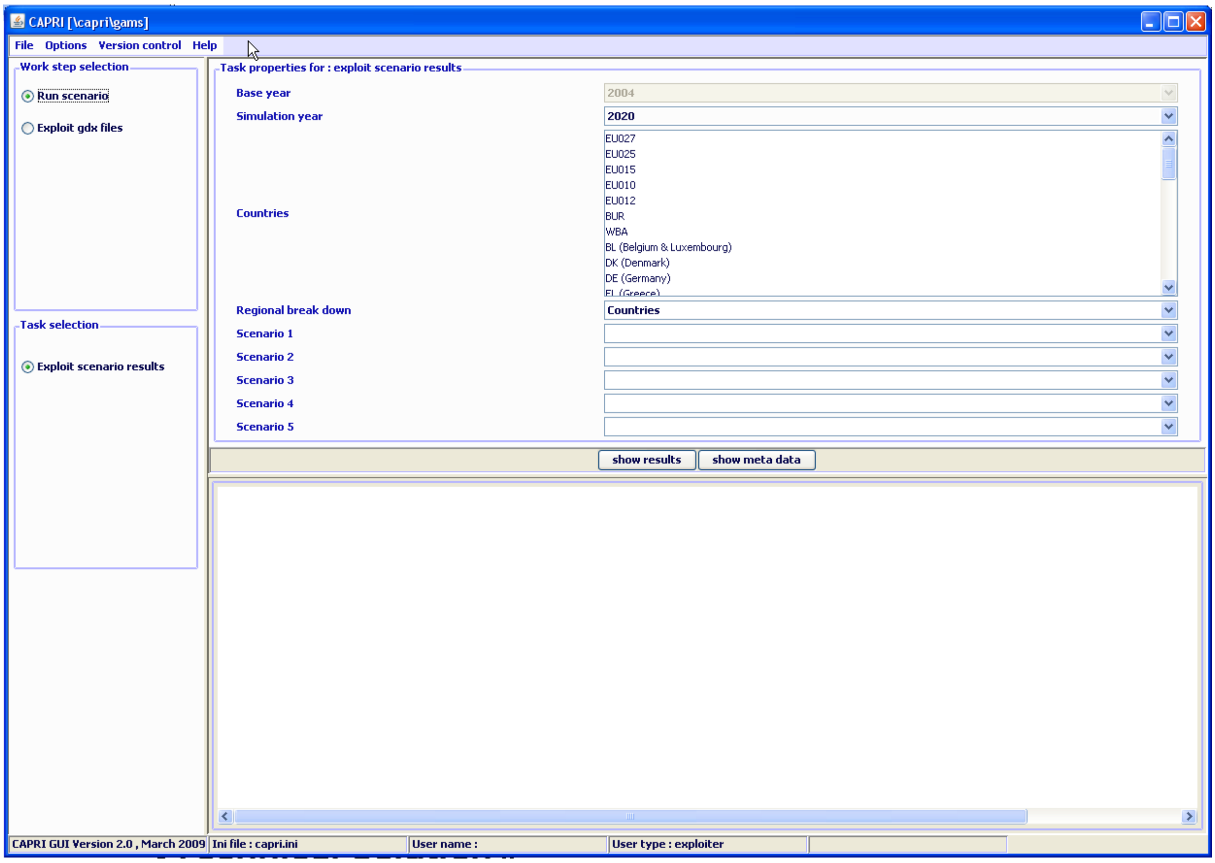
The interface is set-up such that only the results of those work steps are visible where result files are found. For a training session concentrating on analysing scenarios, only those result files can be distributed. An installation with four scenarios at NUTS2 level plus all the necessary GUI files will require under 100 MByte disk space.



Result Directory d:\TSCheltenham2010\results

□ Rename 'capmod.lst' etc. by 'task+settings.lst'

Once the user has optionally entered the results directory, and stored it to the ini file, the user will face a rather clean interface which only allows to exploit existing scenarios and to exploit GDX files (also that option could be removed for exploiters).



● Run scenario

○ Exploit gdx files

● Exploit scenario results

Base year	2004
Simulation year	2020
Countries	EU027 EU025 EU015 EU010 EU012 BUR WBA BL (Belgium & Luxembourg) DK (Denmark) DE (Germany) EL (Greece)
Regional break down	Countries
Scenario 1	
Scenario 2	
Scenario 3	
Scenario 4	
Scenario 5	

CAPRI GUI Version 2.0 , March 2009 | Ini file : capri.ini | User name : | User type : exploiter

Technical solution

The software implementation in Java is based on the SVNKIT (http://svnkit.com/) and implemented in the package de.capri.svnkit.