

SwingScaViewer

V1.3.5x

Installation Guide
User Guide

REF : SwingScaViewer/JLP/2012/10

Author : Jean-Louis PASTUREL

Description of Successive Versions

VERSION	DATES	State	Author
V1.0	30/May/2012	Initialisation Version	JL PASTUREL
V1.1.0	15/Aug/2012	Version 1.1.0	JL PASTUREL
V1.1.2	27/Aug/2012	Version 1.1.2	JL PASTUREL
V1.1.8	03/Sep/2012	Version 1.1.8	JL PASTUREL
V1.2.0	09/Sep/2012	Version 1.2.0	JL PASTUREL
V1.3.3	01/Oct/12	Version 1.3.3	JL PASTUREL
V1.3.9	26/Oct/2012	Update for Version 1.3.9	JL PASTUREL
V1.3.18	11/Feb/2013	Update for Version 1.3.18	JL PASTUREL
V1.3.51	06/Dec/2014	Partial Update for version 1.3.51 (PHP-TOOLS)	JL PASTUREL
V1.3.52	18/Dec/2014	Complete documentation PHP-TOOLS	JL PASTUREL
V1.3.53	19/Dec/2014	PHP-TOOLS, use registered functions instead of append file. Correct this documentation	JL PASTUREL

Table des matières

1Introduction.....	5
1.1Context.....	5
2Installation.....	7
2.1Packaging.....	7
2.2Installation of SwingScaViewer.....	7
2.2.1Requirements.....	7
2.2.2Create a deployment directory.....	7
2.2.3De-compaction.....	7
2.2.4Configuration.....	7
3User Guide.....	9
3.1Menu "Files"	11
3.1.1New Project.....	11
3.1.2Open Project.....	12
3.1.3SSH Cnx uploads/downloads.....	12
3.1.4JDBC Requests.....	16
3.1.5Perf with AspectJ/Java Agent.....	21
3.1.6 Add Directory Logs/CSV.....	26
3.2Menu "ScaLogParser".....	28
3.3Menu "ScaFileStats".....	39
3.3.1Sous-Menu "StatDatas"	40
3.3.2Sub-Menu "GenTemplStatDatas" and " LocalTemplStatDatas "	45
3.4Menu "Viewers"	45
3.4.1Sub-Menus "ScaCharting" and "ScaChartingDyn"	46
3.4.2Sub-Menu "ParseAndView"	48
3.4.3"Funny" feature.....	51
3.5Menu "MyCommands".....	52
3.6Menu "Tools".....	53
3.6.1Sub-Menu "TestRegexp"	53
3.6.2Sub-Menu "TranslateDate"	54
3.6.3Sub-Menu "Concat Files"	55
3.6.4Sub-Menu "Horodate Logs"	55
3.6.5Sub-Menu "Sort Lines In File"	57
3.6.6Sub-Menu "Merge Lines"	60
3.6.7Sub-Menu "Hex <=> Dec"	61
3.6.8Sub-Menu "SimpleDateFormat tester"	62
3.6.9Sub-Menu "Clean current csv directory"	62
3.6.10Sub-Menu "Compact all Csv"	62
3.7Menu "PHP -TOOLS".....	63
3.7.1Quick start PHP-TOOLS.....	63
Installation of swingScaViewer (Windows).....	63
3.7.2The different ways at stress time.....	65
3.7.3Configuration for Trace format 1/0 and profiling.....	66
xdebug.ini.....	66

xdebug_header_cookie_scripts.php.....	68
phptools_incl_commun.php.....	68
3.7.4Configuration Xhprof profiler pour swingScaViewer.....	70
xhprof.ini.....	70
xhprof_header_scripts.php.....	70
phptools_incl_commun.php.....	71
xhprof_header_natif.php.....	72
xhprof_fheader_mixed.php.....	73
3.7.5Configuration uprofiler for swingScaViewer.....	74
uprofiler.ini.....	74
uprofiler_header_scripts.php.....	74
phptools_incl_commun.php.....	75
uprofiler_header_natif.php.....	76
uprofiler_header_mixed.php.....	78
3.7.6Screens shots.....	79
The Xdebug Trace with Format 1.....	79
Sub-Menu XDebug_profile.....	83
Sub-Menu Xhprof.....	85
3.8Menu "ScaViewer Infos".....	89
4General Procedure.....	89
5Annexe.....	94
5.1Examples of Java/Perl regex.....	94
5.1.1Regexp : ^\d+\.\d+\.\d+\.\d+.....	94
5.1.2Regexp : \[[^\]]+\]\.....	94
5.1.3Regexp : "[^"]+"\\s+\\d{3}\....	95
5.1.4Regexp : \\s+\\d+\\$.....	95
5.2Use of key word “function” in swingScaViewer/scaLogParser.....	95
5.2.1File to treat.....	95
5.2.2Classes Scala "myPlugins".....	96
5.2.3Configuration of swingScaViever/ScaLogParser.....	97
5.2.4Visualization.....	99

Advertisement :

English is not my native language, so in the document, you will find a lot of syntax and grammar mistakes, awkwardness, difficulties to understand some sentences ...

*Please let me know at jean-louis.pasturel-wrong-reply@orange.fr
(Remove -wrong-reply to the mail address)*

1 Introduction

1.1 Context

Important :

To use this tool, you must know basics on Pattern Matching with regular expression (Perl regex for example). At the end of the document, I give some examples of regex that are currently used in **swingScaViewer**.

The general mechanism used in this tool is to parse in 2 phases :

- First phase : first regex extracts from a source that contains the interesting information in a result
- Second phase, **if necessary** : second regex extracts, from precedent result, the final information

This mechanism can handle almost all cases

The product **SwingScaViewer** is a kind of workbench that groups several tools :

- parsing dated logs (system logs, Web servers, WAS, application ...) and converting them into csv files
- visualisation of csv files, or direct visualisation with certain types of logs (JVM GC logs for example).
- Integration of others tools like AspectPerf(packaging AspectJ LTW Weaving for profiling Java application), JDBC Requests, statFilesAdvanced ...
- upload and download of files
- Utilities tools => date <=> dateInMillis, aggregation of files, regex expression tester ...

The tool SwingScaViewer is developed in Scala (2.9.2) and uses several Java API:

SwingScaViewer Core : Apache 2 License <http://www.apache.org/licenses/LICENSE-2.0.html>

Scala License : <http://www.scala-lang.org/print/146>

JFreeChart License : LGPL V2.1 <http://www.gnu.org/licenses/lgpl-2.1.html>

jlpApis.jar (My own Apis) : LGPL V2.1 <http://www.gnu.org/licenses/lgpl-2.1.html>

Jsch (BSD License) : <http://www.jcraft.com/jsch/LICENSE.txt>

akka Actors Apache 2 License : <http://www.apache.org/licenses/LICENSE-2.0.html>

Jcommon : LGPL V2.1 <http://www.gnu.org/licenses/lgpl-2.1.html>

commons-math;1.2 : Apache 2 License : <http://commons.apache.org/math/license.html>

joda-time Apache 2 : <http://joda-time.sourceforge.net/license.html>

jtds Licence : <http://www.kxcad.net/esteco/modeFRONTIER320/legal/jtds/license.html>

AspectJ (Eclipse Project) : Common Public License <http://eclipse.org/legal/cpl-v10.html>

mysql-connector-java GPL License : <http://www.gnu.org/licenses/gpl.html>

ojdbc14.jar : OTN License <http://www.oracle.com/technetwork/licenses/distribution-license-152002.html>

postgresql JDBC Driver : BSD License <http://jdbc.postgresql.org/license.html>

Important :

The available packaging contains only the libraries that have the following licenses : LGPL, CPL, Apache, BSD

The libraries mysql-connector-java and ojdbc14.jar are not included in the free packaging. They have to be downloaded from the Web (only useful for JDBC Requests tool)

ojbcxx.jar => <http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html>

mysql-connector-java => <http://dev.mysql.com/downloads/connector/j/>

Put them in the libExt directory and adapt the classpath in file scripts/swingScaViewer.cmd with the version of this JDBC drivers.

2 Installation

2.1 Packaging

The packaging is done in a form of zip file **SwingScaViewer.zip** which the root is **swingScaViewer**.

2.2 Installation of SwingScaViewer

2.2.1 Requirements

A Sun JDK /Open JDK 1.6.0+ must be installed on your desktop.

2.2.2 Create a deployment directory

For all the document, we suppose that you use a Windows System (there is no difficulty to adapt for a Linux box) and that the installation directory is **c:\opt**.
It is not mandatory to create a directory **swingScaViewer**.

2.2.3 De-compaction

After having downloaded **SwingScaViewer.zip** in **c:\opt** le de-compact-it in this directory.

2.2.4 Configuration

The configuration is set in the start script of **SwingScaViewer** (it may be change with new version, but the principle will be conserved) :

File **c:\opt\swingScaViewer\scripts\swingScaViewer.cmd**

```
set root=C:\opt\swingScaViewer
set workspace=C:\opt\workspaceLP
set CLASSPATH=%root%\scaViewer.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\jcommon-1.0.17.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\jfreechart-1.0.14.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\scala-dbc.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\scala-library.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\scala-swing.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\commons-math-1.2.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\jlpApis.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\jsch-0.1.46.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\jtds-1.2.2.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\myaspectjweaver.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\mysql-connector-java-5.1.7-bin.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\ojdbc14.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\postgresql-8.3-604.jdbc3.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\springmyaspectjweaver.jar
set CLASSPATH=%CLASSPATH%;%root%\libExt\akka-actor-2.0.3.jar
set CLASSPATH=%CLASSPATH%;%root%\config
set CLASSPATH=%CLASSPATH%;%root%\myPlugins
set CLASSPATH=%CLASSPATH%;%root%\myPlugins\myPlugins.jar
Set GC_OPTS=-server -Xms1024M -Xmx1024M -XX:NewRatio=2 -XX:+UseParallelGC -XX:MaxPermSize=512M
-XX:PermSize=512M -XX:-UseAdaptiveSizePolicy
REM Set GC_LOGS=-verbose:gc -XX:+PrintGC -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:
+PrintGCTimeStamps -Xloggc:%root%\logs\GC.logs
```

```
java %GC_OPTS% %GC_LOGS% -Dworkspace=%workspace% -Droot=%root% -Dconfig.file=%root%\config\scaViewer.properties -Xms1024M -Xmx1024M com.jlp.scaviewer.ui.SwingScaViewer
```

At the beginning of the file (in bold characters), 2 environment variables must be set, according to your installation.

The Window Environment Variable PATH has to make reachable the executable **java** of the JVM.

Important : The **manual creation** of the directory pointed by **%workspace%** is **mandatory**.

We can after create a link **Windows** on the desktop to launch **swingScaViewer** by a click on the mouse.

3 User Guide

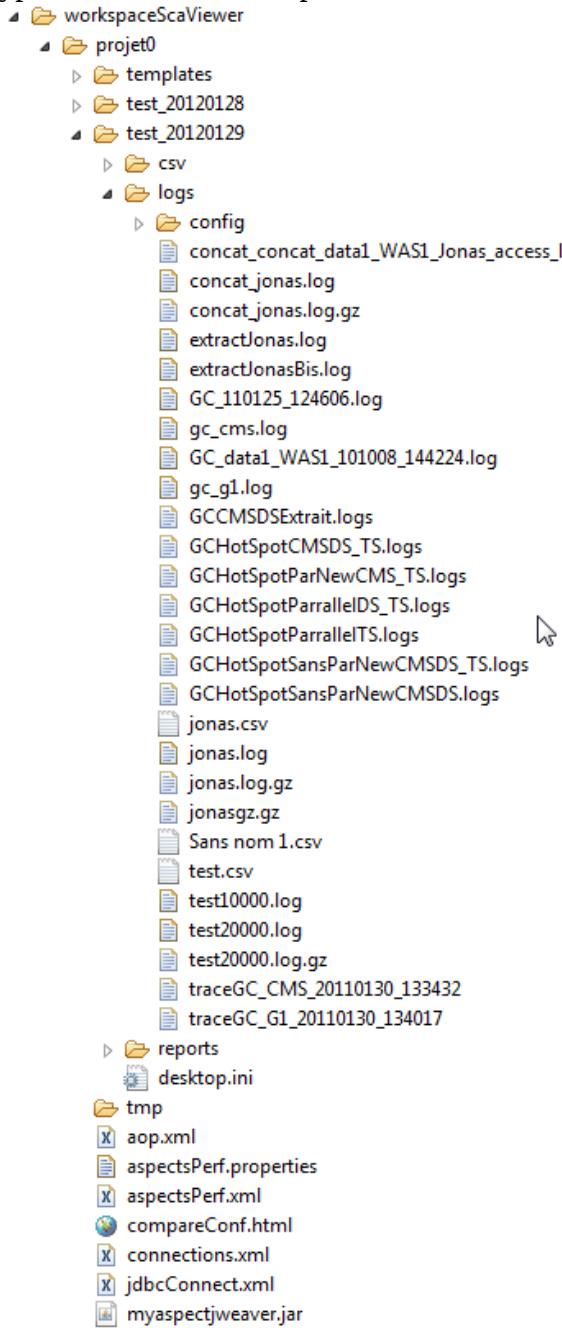
For every feature of **swingScaViewer**, we are going to describe the threads of the screens and the parameters to set in the corresponding screens.

The screen-shots shown in this document correspond with the Version 1.x of **swingScaViewer**. The general principle is to organize the **%workspace%** directory by project and by sub-folders. These last sub-folders may be created by date, by type of scenario or other sorting keys.

When the scenario sub-folders is created, it is automatically prefixed by the variable (if not yet correctly prefixed) : **scaviewer.prefixscenario** initialised in the file :
config/scaViewer.properties

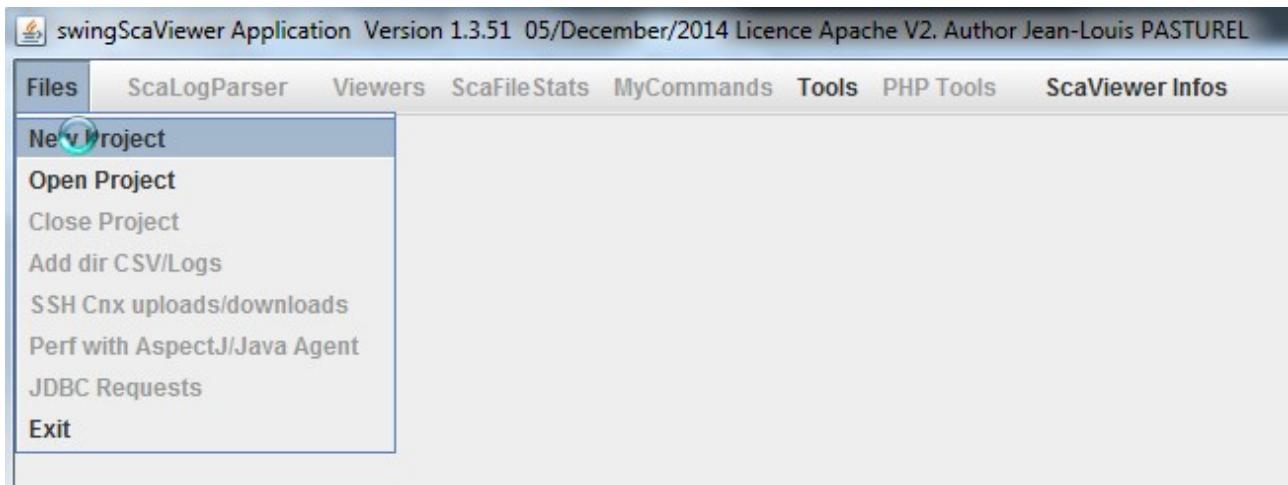
For the rest of the document, these sub-folders will be named “scenario folder”.

A typical tree in the workspace folder looks like this:



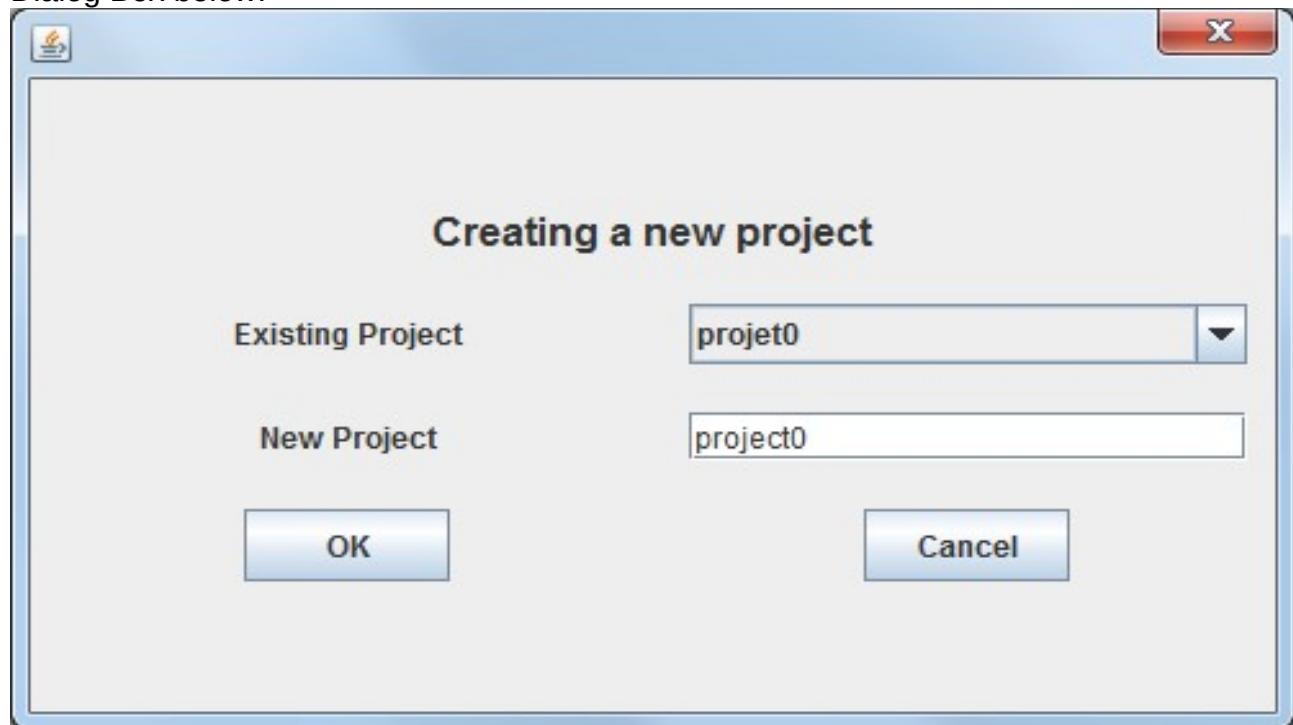
In the **logs** folder, there are the original access-logs or application logs files and in the **csv** folder, the csv files generated after treatment of logs files.

3.1 Menu "Files"



3.1.1 New Project

The sub-menu **New Project** allows to create a new project, which name is set in the Dialog Box below:



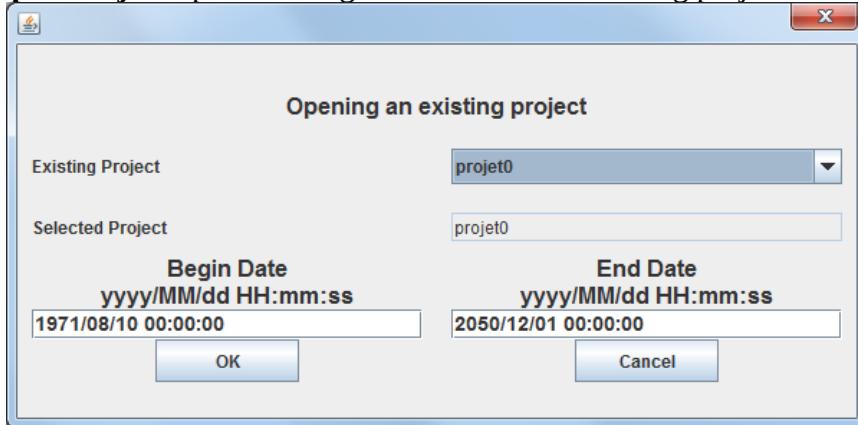
Click OK and a new directory **c:\opt\workspaceScaViewer\project0** is created if **%workspace%**

is set to **c:\opt\workspaceScaViewer**

This operation must be the first operation for a new project.

3.1.2 Open Project

The sub-menu **Open Project** open a Dialog box to choose an existing project:



We can choose for example the existing project **projeto**

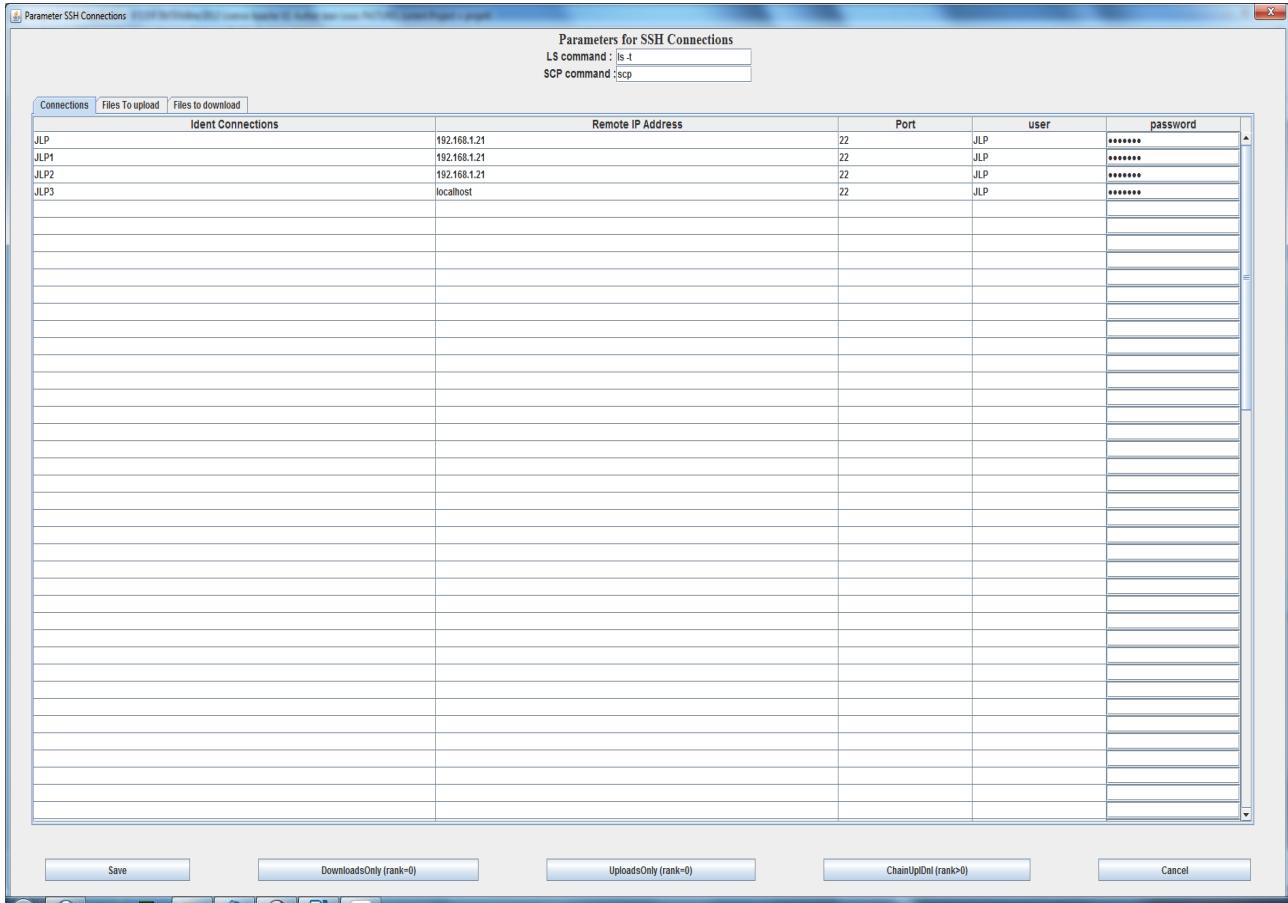
Starting with version 1.2, there is a begin / end date for the current scenario. The fields can be updated. It increases performance when parsing large files, which the logs are not all useful.

Note : By creating or opening a project, we enable menus on the MenuBar. Closing the project disables these same menus.

3.1.3 SSH Cnx uploads/downloads

The sub-menu **SSH Cnx uploads/downloads** allows to automatize the downloads of log files or entire directories from the servers to the desktop that runs **swingScaViewer**

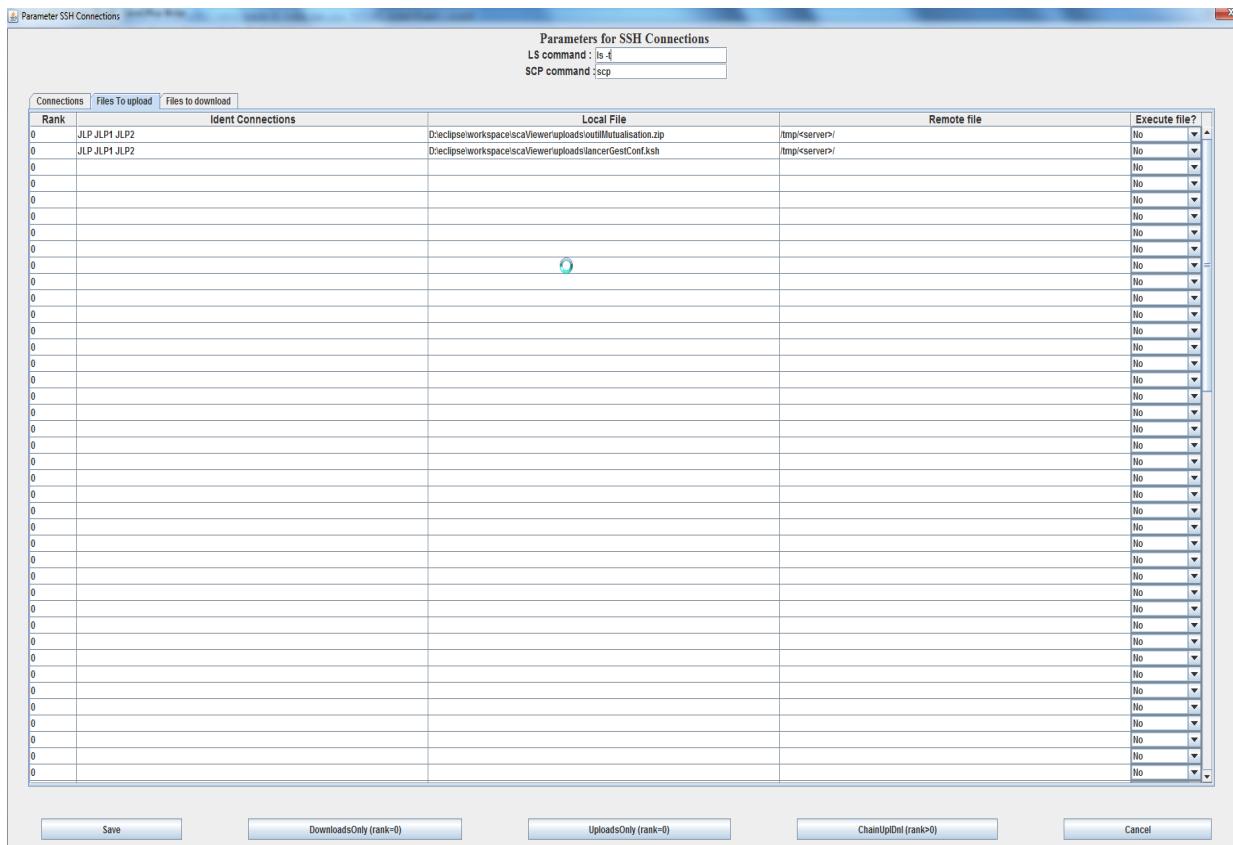
This mechanism is possible with direct ssh or throw ssh tunnels which are before hand opened.



The first tabPanel allows to define the connections (ident Servers, Ip Address, ssh port, login, password).

Hint : It is allowed to fill the same server with different names in the first column : it is useful when the files in a server have the same names (case of multiple WAS for example on the same server logs with the same names), the download mechanism in **swingScaViewer** prefix the file with the name of the server.

We can also adapt the command for listing files and also the command of scp. The default configuration fits well with Linux and Aix servers, no change is needed.



The second tab panel, allows to uploads files to the server (as for examples shell scripts)
The first column contains the rank of the operation :

- 0 means that the operation is launched when **UploadsOnly** button is clicked
- -1 means that the line is skipped and not executed.
- > 0 means that this upload is a part of a series of upload/download with this rank
(executed only when clicked on **ChainUpDnl** button)

The second column contains the idents of the connections.

For this second, you can use several pattern :

- **allip** => upload files/directories to all different ip addresses defined in the tab connexions
- ***** => upload files/directories to all different servers defined in the tab connexions
- **server1 server2** => a list separated by a space character upload to the servers on the list
- **serv*** => upload to the servers which the name begins by serv

The third column contains the file to upload (by right clicking on this column, there is a File chooser that points to the uploads folders that contains pre-defined shells).

Also the pattern **<server>** is usable.

The fourth column contains the name of the folder of the remote server where the file will be uploaded (the last character must be the file separator as for example **/tmp/**)

The fifth column indicates if you want to execute the shell, after it was be uploaded.

Note : Before downloading files or directory, verify that you have created a new sub-folder i scenario by the Menu : **Add dir CSV/Logs** (see further in the document)

The third tab-panel allows to define the files or the directories that we want to download.

The first column (rank) has the same behaviour of the Upload tab:

The first column contains the rank of the operation :

- 0 means that the operation is launched when **DownloadsOnly** button is clicked

- -1 means that the line is skipped and not executed.
- > 0 means that this upload is a part of a series of upload/download with this rank
(executed only when clicked on ChainUpIDnl button)

The second column is the key of a server (corresponding to the first column of the first tab-panel)

For this first column, you can use several pattern :

- **allip** => download files/directories from all different ip addresses defined in the tab connexions
- ***** => download files/directories from all different servers defined in the tab connexions
- **server1 server2** => a list separated, by a space character, downloads from the servers on the list
- **serv*** => download from the servers which the name begins by serv

The third column sets the type of object to download (Files, FilesNoPrefix, Directory, Explicit_Cmd).

- **Explicit_Cmd** is used when you have to choose files that can't be reached by ls (example with command find to get files with a minimum size or located in different branches of a directory)
- **FilesNoPrefix** doesn't prefix the downloaded file by the name of the server

The fourth column set the name of files or directories to download. The joker character * is now allowed everywhere in the string (beginning with version 1.1.2). You can use also the pattern <server> in the path of the file on the path of the directory, it will be replaced by the corresponding name of the server.

The fifth column indicates how many most young objects we want to download.

The sixth column is the choice of the target folder where the download will occur (logs in directory logs of the scenario, csv logs in the csv directory of the scenario, reports in the reports directory)

The seventh column indicates if you want to gzip the file before downloading

3.1.4 JDBC Requests

The sub-menu **JDBC Requests** allows to periodically request databases of an application through a JDBC connections (When it is possible, Firewall or database listener can forbid the access)

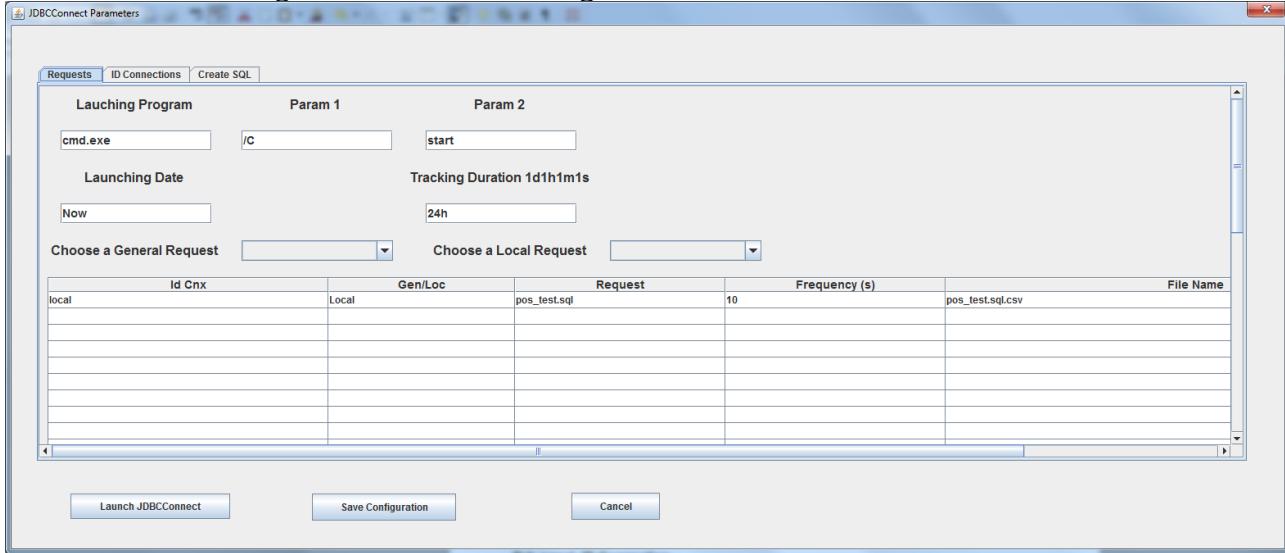
The supported databases are :

- Oracle
- Sybase
- SQL Server
- MySql
- PostgreSQL

As said, more above, the Oracle JDBC Driver and MySql JDBC Driver must be downloaded from

the Web and put in the libExt directory.

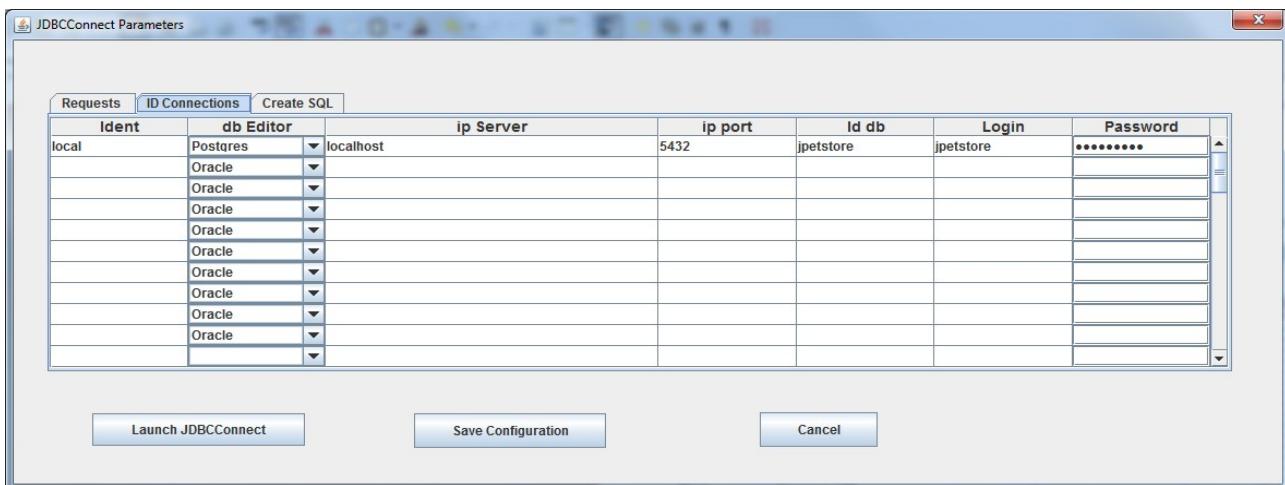
The screen that configure JDBCConnect are given below :



In this tab-panel, we set the parameters to launch JDBC Requests, that can also run after stopping **swingScaViewer**.

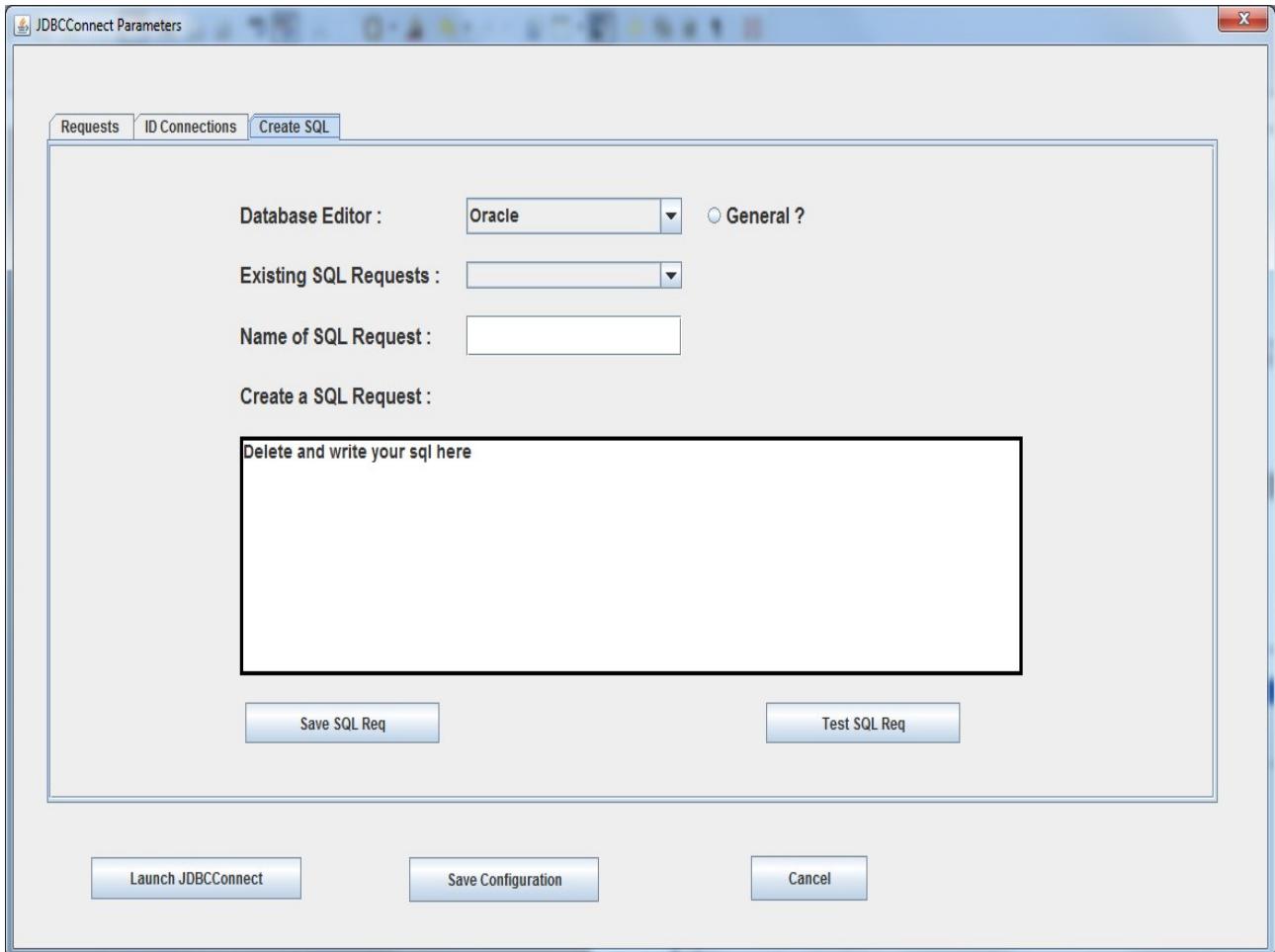
The two combo-boxes allow to chose existing requests (see also how to create a SQL request below). The requests are send to the IDConnexion (see the corresponding tab-panel).

Tab-panel ID Connection :



In this tab-panel, the parameters needed by a JDBC connection must be set

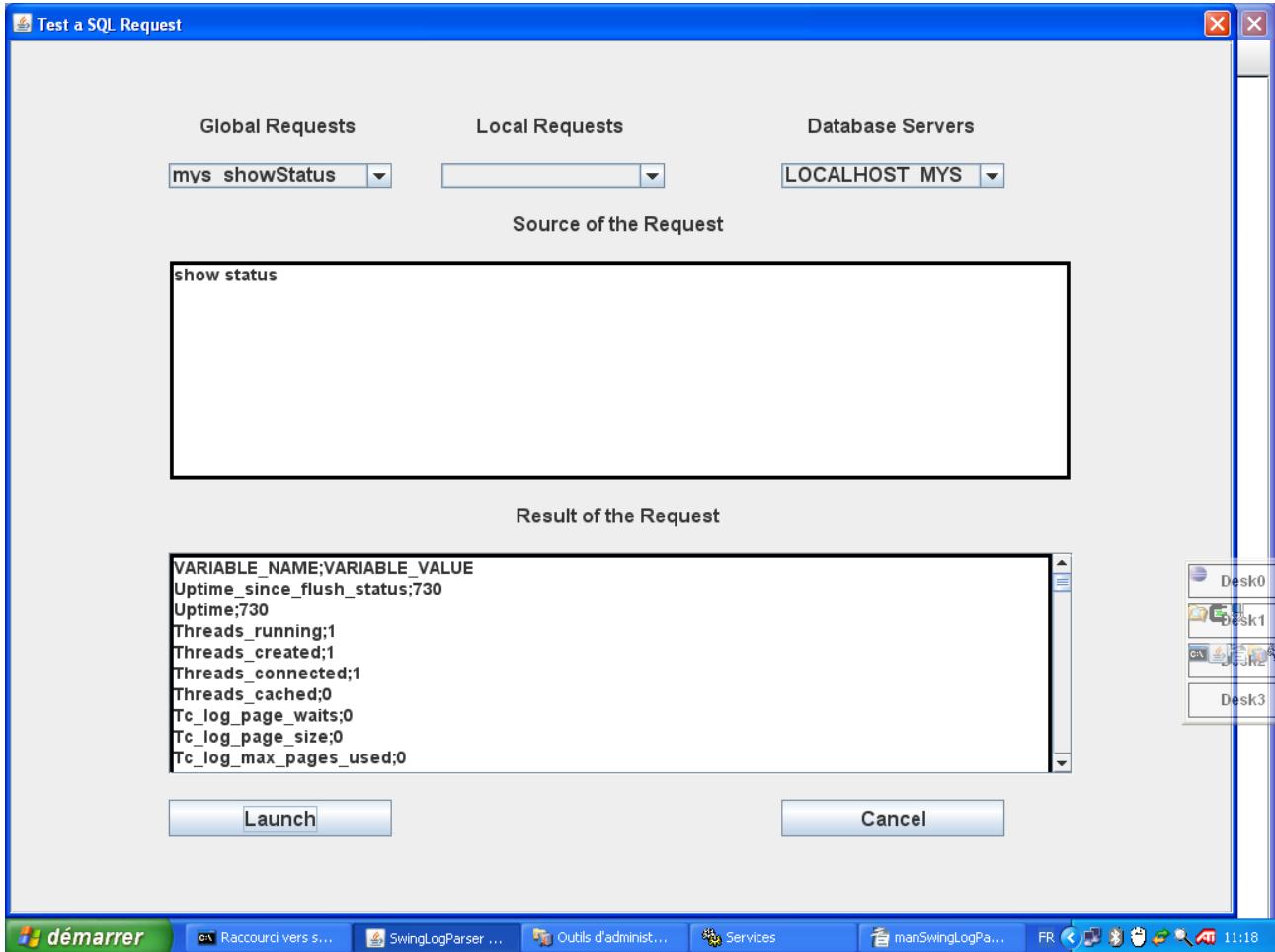
Tab-Panel Create SQL



This tab-Panel allow to create and test SQL requests.

In the above example, the request **show status** toward a MySQL database gives a complete state of the database.

If we do a unit test, for this request, we obtain the result shown below :



The launch of JDBC Requests with the parameters above gives the Windows Console command below :

```
JDBCConnect Projet = monProjet
JJDBCConnect launched at : 2009/Jan/04 11:19:50
JJDBCConnect launched with parameters :C:\opt\workspaceLP\monProjet\jdbcConnect.xml
Le fichier de configuration JDBC existe, on remplit
JJDBCConnect : Collect will begin at 2009/01/04:11:19:50, for a duration of 24h30m
Before launching threads
Starting thread : LOCALHOST_MYS/mys_showStatus
After launching threads
Starting thread : LOCALHOST_MYS2/mys_showStatus
```

The result files are created in the most recent logs directory. The files are generated in gz format. So you have to kill the Windows cmd console to generate a correct gz final file.

Example with the parameters set above :

```
Time;VARIABLE_NAME;VARIABLE_VALUE
2009/01/04:11:19:51;Uptime_since_flush_status;847
2009/01/04:11:19:51;Uptime;847
2009/01/04:11:19:51;Threads_running;2
2009/01/04:11:19:51;Threads_created;2
2009/01/04:11:19:51;Threads_connected;2
2009/01/04:11:19:51;Threads_cached;0
2009/01/04:11:19:51;Tc_log_page_waits;0
2009/01/04:11:19:51;Tc_log_page_size;0
2009/01/04:11:19:51;Tc_log_max_pages_used;0
2009/01/04:11:19:51;Table_locks_waited;0
2009/01/04:11:19:51;Table_locks_immediate;18
2009/01/04:11:19:51;Ssl_version;
2009/01/04:11:19:51;Ssl_verify_mode;0
2009/01/04:11:19:51;Ssl_verify_depth;0
2009/01/04:11:19:51;Ssl_used_session_cache_entries;0
2009/01/04:11:19:51;Ssl_sessions_reused;0
2009/01/04:11:19:51;Ssl_session_cache_timeouts;0
2009/01/04:11:19:51;Ssl_session_cache_size;0
2009/01/04:11:19:51;Ssl_session_cache_overflows;0
2009/01/04:11:19:51;Ssl_session_cache_mode;NONE
2009/01/04:11:19:51;Ssl_session_cache_misses;0
2009/01/04:11:19:51;Ssl_session_cache_hits;0
2009/01/04:11:19:51;Ssl_finished_connects;0
```

The dates in the file allow to directly chart it with the Menu Viewers (see further in the document)

3.1.5 Perf with AspectJ/Java Agent

Advanced use for profiling JAVA application

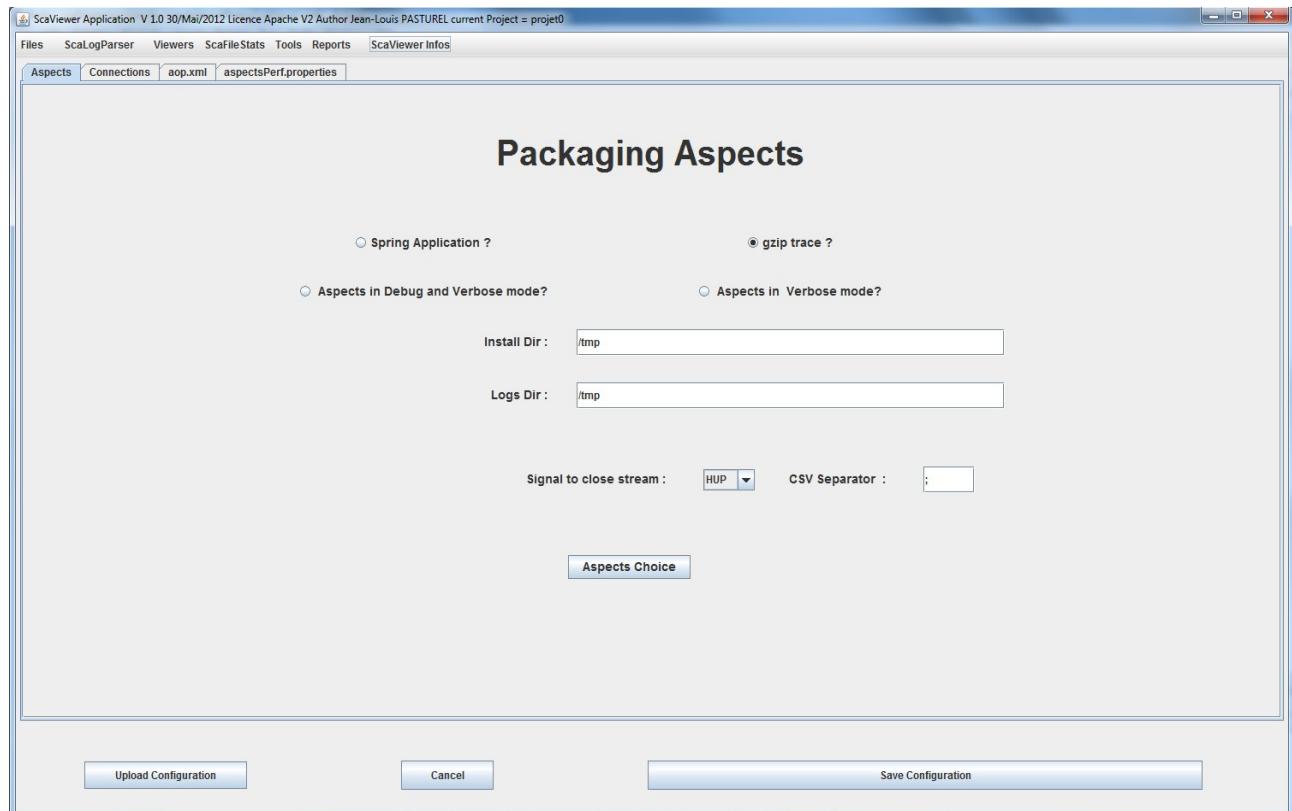
The sub-menu **Perf with AspectJ/Java Agent** allow to package in a single jar file an Aspectj java agent (**myaspectjweaver.jar**) that contains:

- the Load Time Weaver **aspectjweaver** (LTW) standard of Eclipse project AspectJ
- The Spring-agent **spring-agent.jar** (LTW) used for certain forms of weaving with Spring (it is recommended to read the Spring AOP docs before)
- The collection of Aspectjs of the tool **AspectsPref**
- The standard configuration file **aop.xml** of AspectJ
- the configuration file **aspectsPerf.properties** of the **AspectsPerf** tool.

The interest to make a single jar is that the aspectj are in in the classpath at the start of the JVM. Some Aspects may need to add some other jar at the bootclasspath. For example to profile Jolt, Axis, CXF ...

The tool allows, when the packaging is ready to upload it to the remote server.

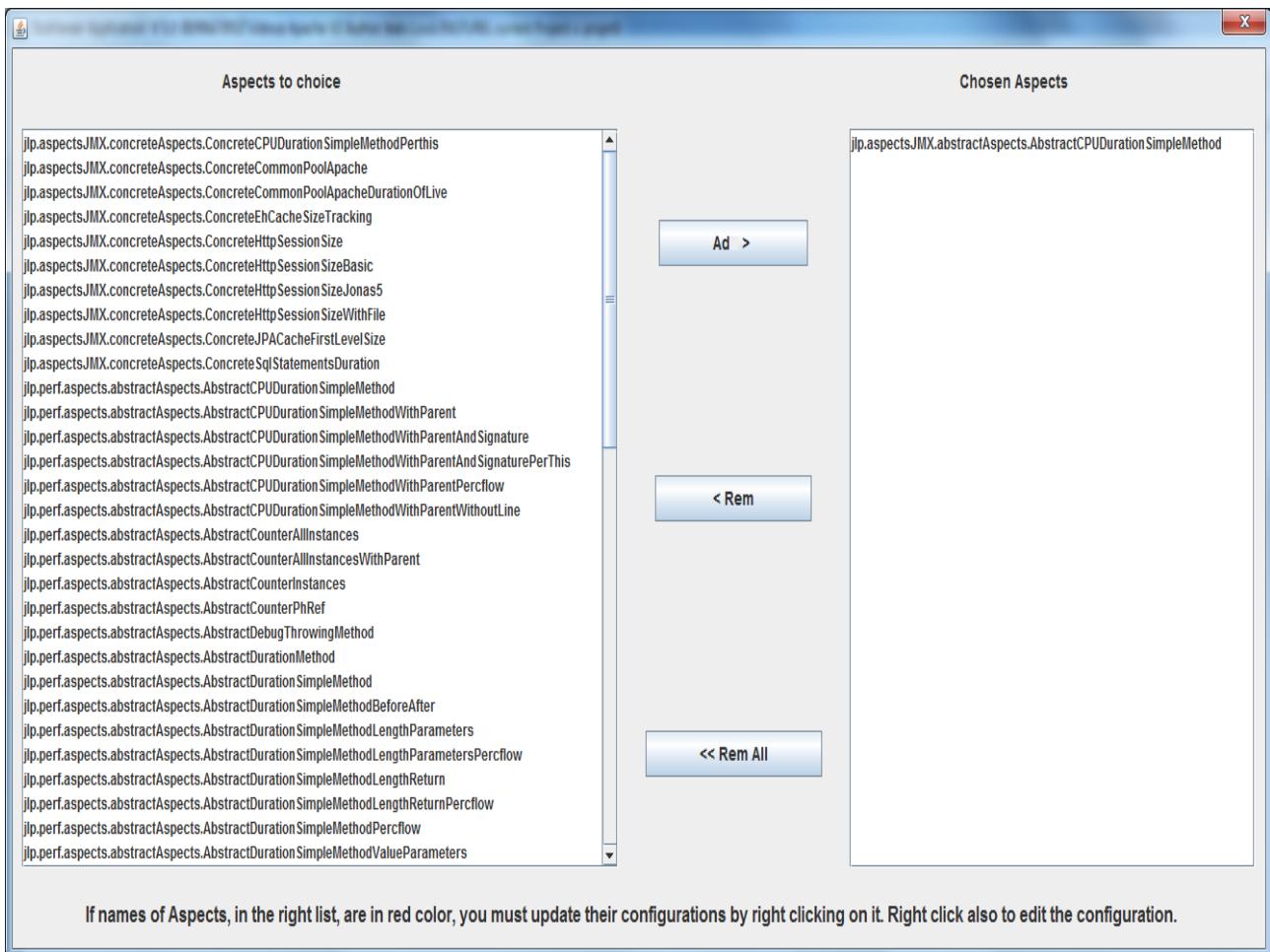
The screen involved in this packager are given below :



The tab-panel Aspects defines :

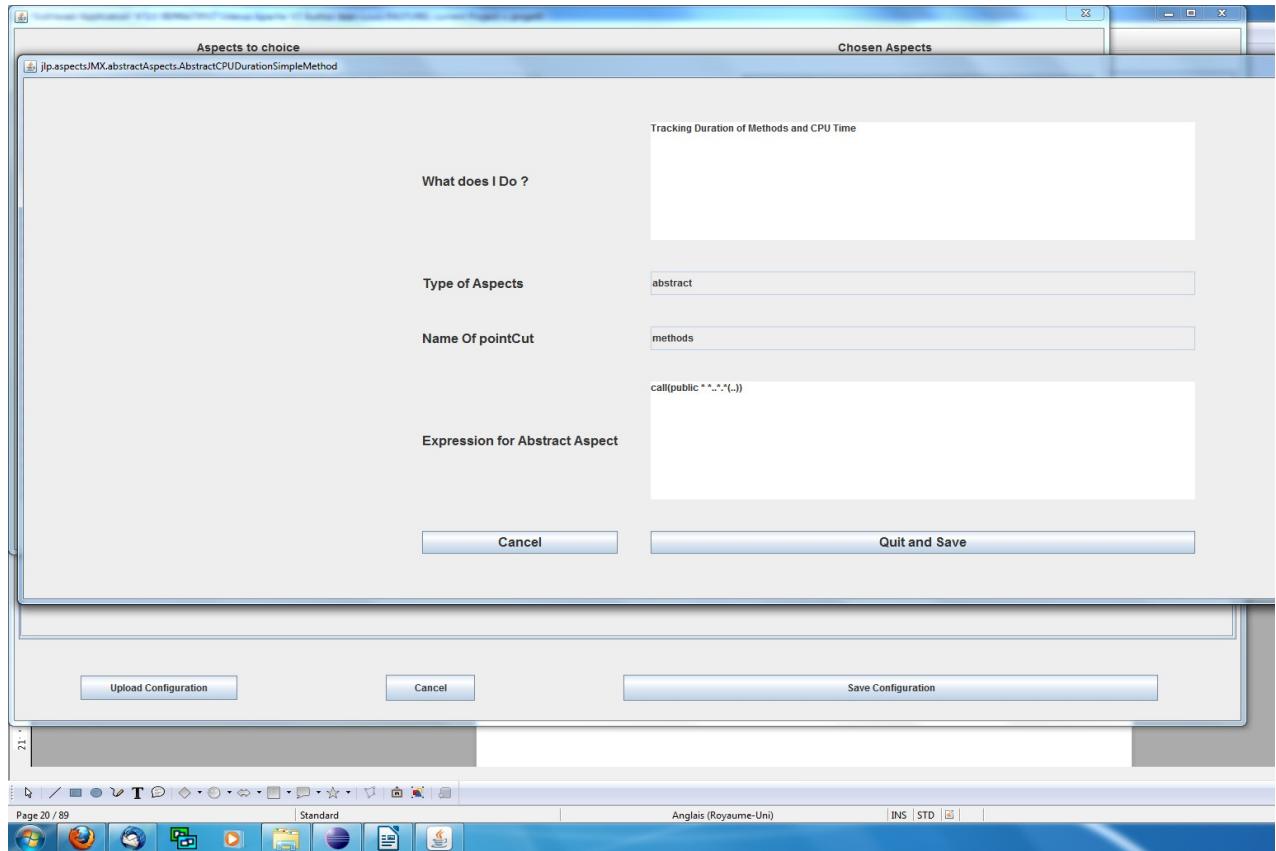
- what kind of agent weaver to use (Standard AspectJ or Spring, standard advised)
- the gzipped or not of the logs generated
- The debug/verbose mode of Aspectj

- the directory of the remote server where to upload myaspectjweaver.jar
- the directory of remote server where the logs will be generated
- the system signal (Unix only, HUP advised) to close correctly the gzip file and rotate the logs
- A Button that allows to choose an Aspect between a collection that can be extended
- Below different parameters linked at the aspect :
- The definition of the pointcut for abstract Aspectj, that define the scope (package/classes) on which the Aspect is woven. The docs of AspectJ explains all the syntax of the pointcut. (<http://www.eclipse.org/aspectj/doc/released/progguide/starting-aspectj.html#pointcuts>)



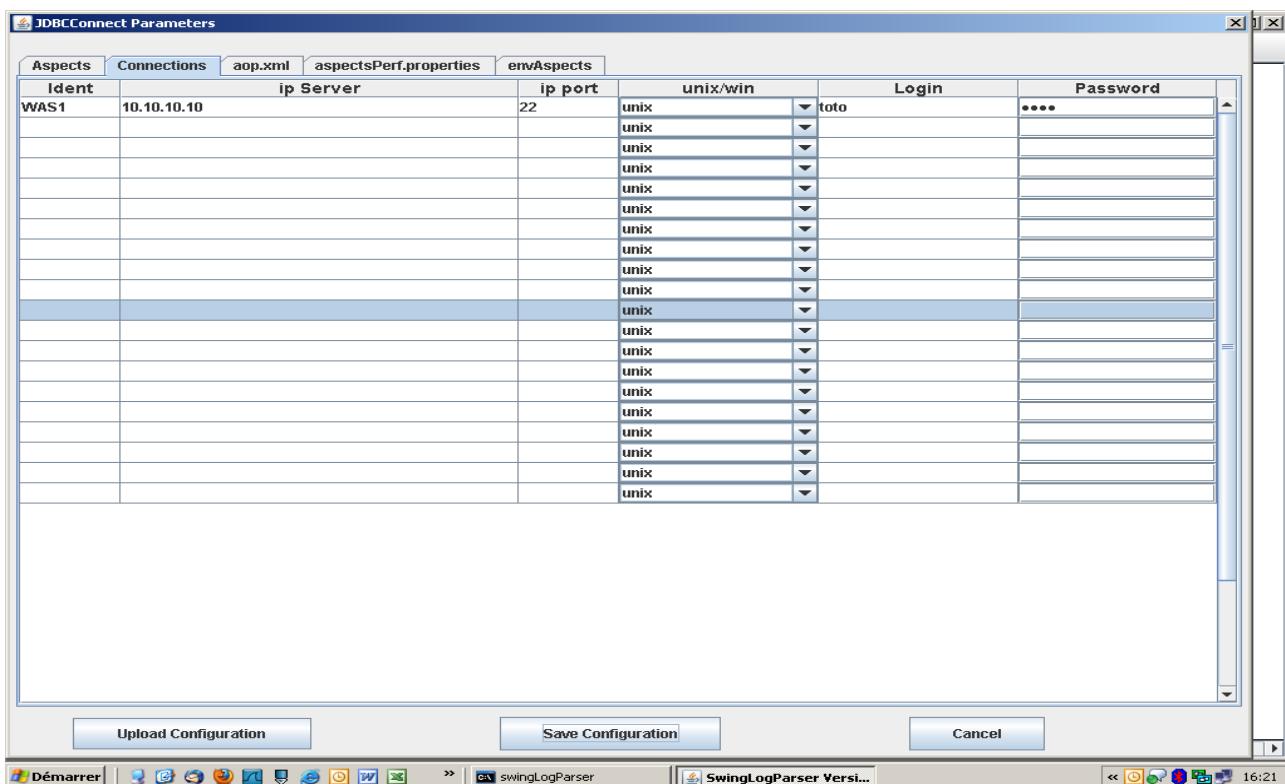
The Panel before allow to choose one or more Aspects.

For each aspect, by right clicking in a selected aspect, you can configure it as shown below :



Every event on the button **Quit and Save** modifies the file **aop.xml**, located on the third tab-panel.

Tab-Panel Connections :



This tab-panel defines on which servers the agent must be uploaded

Tab-Panel aop.xml :

```

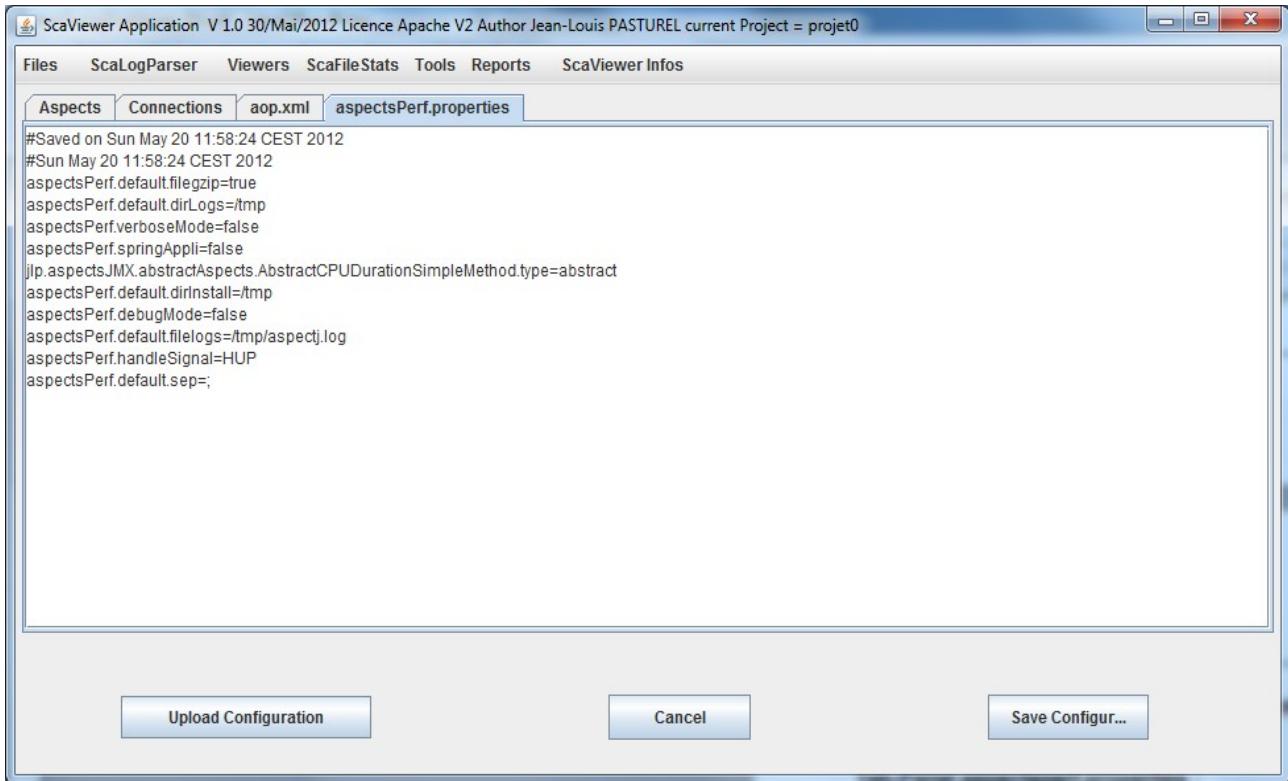
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?><aspectj>
<aspects>
    <concrete-aspect extends="jlpapectsJMX.abstractAspects.AbstractCPUDurationSimpleMethod" name="jlpapectsJMX.abstractAspects.AbstractCPUDurationSimpleMethod">
        <pointcut expression="call(public * *.*(..))" name="methods"/>
    </concrete-aspect>
</aspects>

<!-- Foot Mode Normal -->
<weaver options="-Xlint:ignore -Xset:overWeaving=true,typeDemotion=true,weaveJavaPackages=true,weaveJavaX Packages=true">
    <exclude within="jlpaperf..*"/>
    <exclude within="org.ow2.util..*"/>
    <exclude within="jlpapectsJMX..*"/>
    <exclude within="jlpapects..*"/>
    <exclude within="org.hibernate.type.ComponentType"/>
    <exclude within="*.EnhancerByCGLIB..*"/>
    <exclude within="*.*$EnhancerByCGLIB$*"/>
    <exclude within="*.*$FastClassByCGLIB$*"/>
    <exclude within="*.*$Proxy*"/>
    <exclude within="$Proxy*"/>
    <exclude within="org.apache.juli..*"/>
</weaver>
</aspectj>

```

This tab-panel shows the contents of the file **aop.xml**. (read-Only access, modifications are not saved)

Tab-Panel **aspectsperf.properties** :



This tab-panel shows the contents of the file **aspectsperf.properties**. (read-Only access, modifications are not saved)

Note :

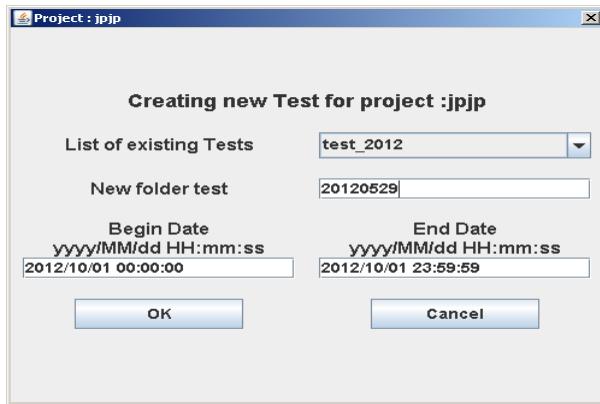
The library of Aspects is extensible, but I have not an available documentation. If somebody is interested by the source code of AspectPerf, I can send it by mail. My mail address is given at the beginning of this documentation

3.1.6 Add Directory Logs/CSV

The sub-menu **Add Directory Logs/CSV** allows to structure the directory of the project in scenarios or date of tests.

The scenario or test name is what you type in the Dialog Box below.

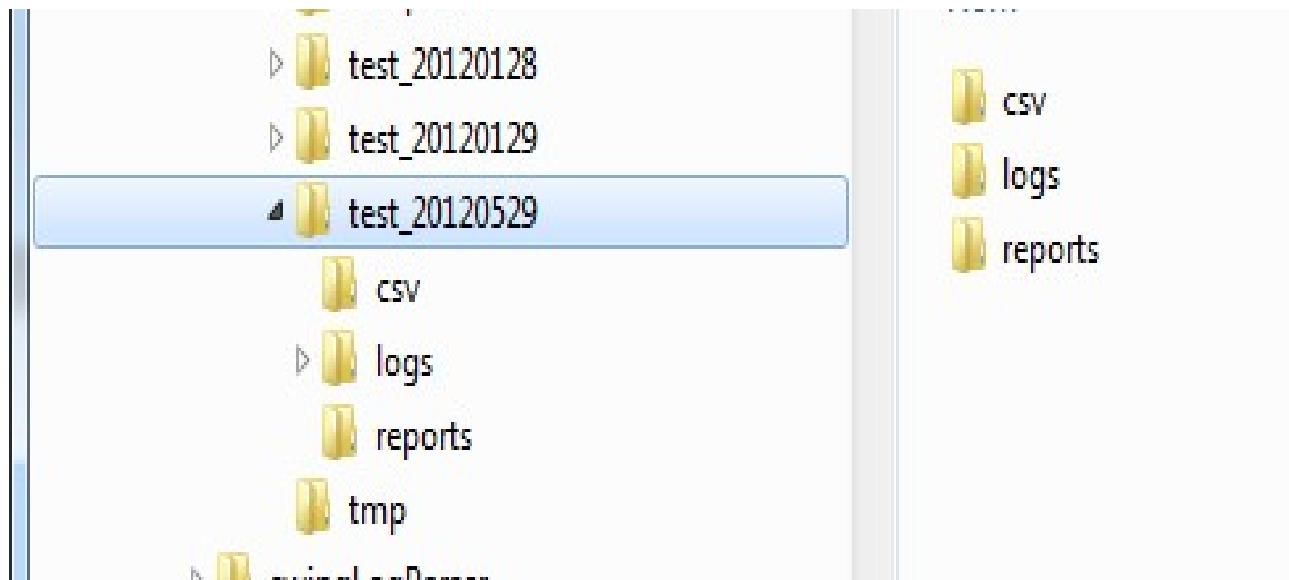
Important : When you create a New Project, you must also create a first scenario.



the new directory is created as shown below :

%workspace%\project\test_20120529

Starting with version 1.2, there is a begin / end date for the current scenario. The fields can be updated. It increases performance when parsing large files, which the the logs are not all useful.



In the sub-directory **logs** of the new folder (**test_20120529**), we put all logs files corresponding to the new scenario . As see more above, the automatic download help you to handle the logs files from the remote server.

Note : to remove a project, remove the directory of the project .

3.2 Menu "ScaLogParser"

With this menu, we transform a log file to a csv file that can be graphed with JFreeChart (Menu Viewers)

The Menu ScaLogParser (in fact a Button) open a customized FileChooser in the directory
%workspace/<Project>/last_scenario/logs

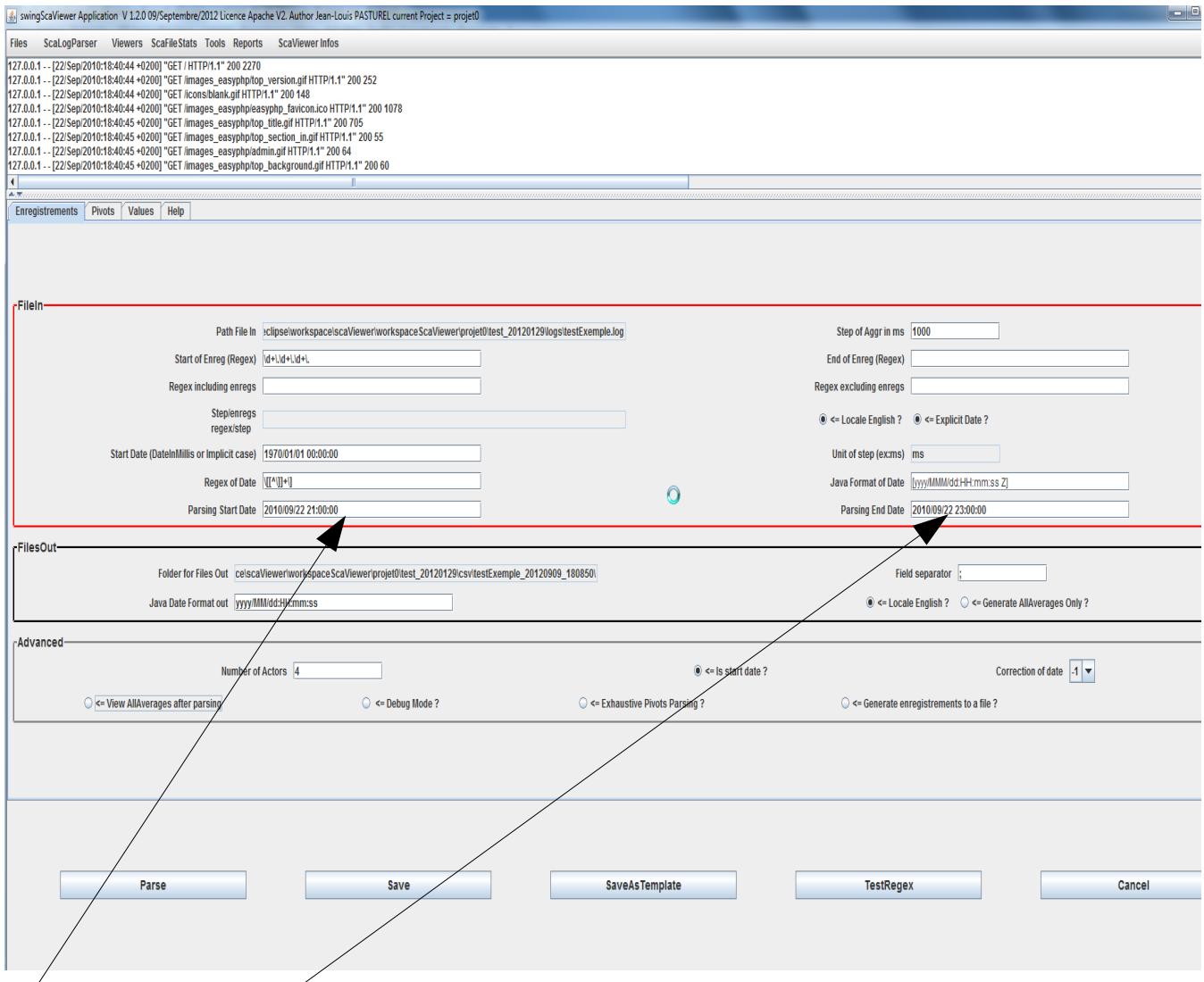
You have to choose a file.

There are 3 possibilities to configure before parsing :

- with no template, (ex-nilho) => ComboBox empty
- with a local template, only valid for the current project General Template unselected and a template chosen in the ComboBox
- with a general template, shared by all projects General Template selected and a template chosen in the ComboBox

The screens below, show the more general configuration with no template

After clicking on OK Button, the screen below appears :



Parsing Start Date / Parsing End Date : new in version 1.2.0 to set a beginning and an end date for parsing. It is useful when log files are large. See the table below for the format.

In the TextArea at the top of the Windows, there are the first 20 lines of the log file to parse

In the bottom part, there is a Tabbed Pane with 4 tabs

The first tab has the functions :

- to define what is a record in the source file (Beginning/End of record, Locale ...)
- to retrieve the date of the record (explicitly or implicitly).
- to define the format of the output CSV file(Date format, separator)
- Automatic graph or not after parsing

File In parameters :

Parameters	Signification	Example
Path File In	Full path of the source file. Automatically filled by swingScaViewer when chosen	
Step of Aggr in ms	Measurement period of result aggregation in milliseconds.	1000
Start of Enreg (Regex)	Regular expression (Perl Regex) to find the beginning of the record. Mandatory .	\d{4}-\d{2}-\d{2}
End of Enreg (Regex)	Regular perl expression to catch the end of a record. If this field is empty, it signifies that each line is a record.	If there is no evident end of record pattern, you can set it the same pattern as the pattern of the beginning of the record. swingScaViewer takes care not to skip any record.
Regex including enregs	Regular perl expression that allows to include all records that are interesting.	Exclude feature has priority over Include feature.
Regex excluding enregs	Regular perl expression that allows to exclude some records that are not interesting.	
Step within enregs or regex for step	Used when Explicit Date is unselected. Define the increment relative to the parameter fileIn.startDate under 2 ways. => regexp a perl regex val , a value in between 2 records (unit is given by Unit of Step)	(?![^=]=)\d+ val=1000 (* See more explanation after this table.)
Locale English	Choice of the Locale for File In (for Date and representation of Double)	Selected (as possible)
Explicit Date	If selected, means that every record of the file contains an explicit date. If not (Implicit) the date must be computed.	Selected (as possible)
Start Date (DateInMillis or Implicit case)	Definition of a beginning date with the Java DateFormat defined by the field Java Format of Date	1970/01/01 00:00:00 (0 ofTimeStamp) Used when Explicit Date is unchecked, if so an increment must be defined. or when Regex of Date is set to dateInMillis with explicit dates.
Unit Of Step	Unit of the increment. Possible values are s,ms,micros,nanos	Ms
Regex of Date	Regular expression (perl regex) to catch the date. The date must be in the first selected group. In case of date in timestamp in millis the pattern to set is dateInMillis .	^(?:[^:]+[:][^:]+) if dateInMillis in java date format , you can put here two regex separated by a blank to extract the date in millis (the 2 regex must not contain a blank)
Java Format of Date	Select the Java DateFormat of the records of	[dd/MMM/yyyy:HH:mm:ss Z]

Parameters	Signification	Example
	the log file.	or dateInMillis or dateInMillis,<mult> mult=1000 if date is in seconds The dateInMillis format, compute date, starting with the beginning date filled in the Start Date (DateInMillis or Implicit case) parameter
Parsing Start Date	The date for the beginning of parsing	Format must be : yyyy/MM/dd HH:mm:ss
Parsing End Date	The date for the end of parsing	Format must be : yyyy/MM/dd HH:mm:ss If beginning date >= end date, there is no control for the date. So with the two dates set by default to 1970/01/01 00:00:00, there is no control of date.

File Out parameters :

Parameters	Signification	Example
Folder for Files Out	Full path of the target directory files. Automatically filled by swingScaViewer when chosen	
Field separator	Separator for the output CSV file	;
Java Date Format out	Format DateFormat Java of the csv generated file	yyyy/MM/dd HH:mm:ss
Locale English	Choice of the Locale for File In (for Date and representation of Double)	Selected (as possible)
Generate AllAverages Only	When selected, only the average of the values parsed are generated on a single file.	The name of the file is allAverages.csv

Advanced parameters :

Parameters	Signification	Example
Numbers of Actors	Number of actors (akka Actors) to treat the log file	2 or nb cores * 2 (if hyperthreaded)
Is start date	If selected, means that the date is the date of the beginning of the request	Selected depends of file logs.

Parameters	Signification	Example
Correction to date	When the first value is expressed as a duration the 3 strategies are : if 0 => no modification of the date if 1 => add the duration to the date if -1 => subtract the duration to the date	Select 0 or 1 or -1
View AllAverages after parsing	If checked, at the end of the parsing, the file allAverages.csv is graphed automatically.	Checked/Unchecked
Debug Mode	For debugging with a little extract of the log file. There is a log trace in the directory <swingScaViewer_Home>/logs	Checked/Unchecked
Exhaustive Pivots Parsing	Useful when 2 pivots have inclusive regex for example GET\s/MyHome/ GET\s/ If not checked, the URLs matching the first regex are not parsed by the second regex	Checked/Unchecked If you are sure that regex are all not inclusive, unchecking is faster for parsing.
Generate Enregistrements to a File	Help also for debugging to see if you correctly parse the record of the log file. Useful when the record are spanned in several lines of the file.	Checked/Unchecked

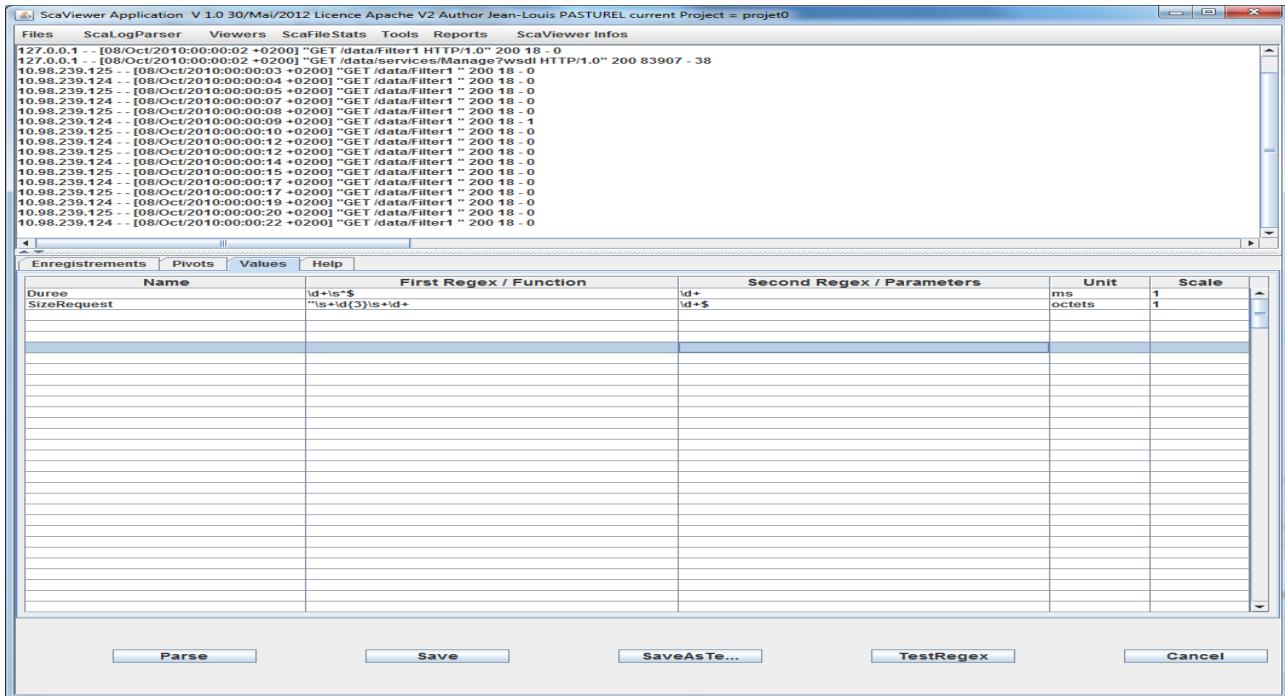
The analysis of access-logs is based on the utilisation of regular expressions that are described in the JDK Javadoc. Look at Pattern class :

<http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html>

(*) The different forms of increment are :

- **<regular expression>**
 - Enter here a regular expression to match the increment. The tool test regexp can help you (see later in this document).
- **val=<Constant value of the increment>**
Specify here the constant value of the increment, the unit is given by the parameter **Unit of Step**. A To be used when there is no explicit date in the access logs and you know the period of generating every record.

Value tab :



The Value tab is divided in 5 columns :

- **Name Value** : in this column you set the name of the value that you want to follow (the name is free)
 - **First Regex / Function** : in this column, you set the mean for extracting the value, there are two ways to extract a value :
 - **regex** (the default shown on the screen-shot above). You put a regex to extract the value. If you are not able to extract the value with this first regexp, you can apply a second regexp(**Second Regex / Parameters**) to extract the final value from the first matching.
 - **function=<nameOfClass>**
 - It is an advanced feature, that allows to write a kind of plug-in to swingScaViewer (useful when the value can't be extracted directly example : sum or difference of 2 values in a record , computing GC throughput...).(**) See below the whole explanation of this feature.
 - **Second Regex / Parameters**
 - A regex when needed as explained above or parameters when using plugins/fuction
 - **Unit :**
 - the unit of the value extracted.
 - **Change Scale :**
 - allows to change the scale of the value to match the unit chosen.

(**) key word function :

It is an advanced topic and it needs to know programming in Scala language.

We must write a **Scala class** with the default package, which has 2 methods with the given

prototypes below :

```
def metInit(tab:Array[String]=null)
def retour(params:Array[String]):Double
```

There are examples in the directory <swingScaViewer_Home>/myPlugins . The classes (byte-code) must be put in the archive jar : <swingScaViewer_Home>/myPlugins/myPlugins.jar. This archive jar is also set in the classpath of the script swingScaViewer.cmd (.ksh).

The return value of the method retour is a **Double** (it can be also Double.NaN, this case is treated).

In the column **First Regex / Function** , the name of the class is set (without the suffix .class) :

function=Add2Values

In the field **Second Regex / Parameters**, we can set all the necessary parameters to execute the function. The parameters are separated by a separator defined in the first character of the field. Be care when there is blank in a parameter to choose another separator than a blank space.

Example :

,param1,parma2,param3

, is the separator, param1 to param3 the parameters.

The record is passed by default to the class, ant it is the params(0) of the parameters of the function **retour**. The others parameters are added at params(1) and so on...

Tip :

When the name of a class begins by Conc, the function can be used in multi-threaded parsing. If one class does not begin by Conc in the Value Tab, the parsing is automatically mono-threaded (The field Nb of Actors is set automatically to 1)

Look at the Annexe part an example of function.

Pivots Tab :

The screenshot shows the ScaViewer Application interface. At the top, there's a menu bar with options like Files, ScaLogParser, Viewers, ScaFileStats, Tools, Reports, and ScaViewer Infos. Below the menu is a list of log entries from Oct 2010, mostly showing GET requests to '/data/filter1' with status 200 and response time 0. The main feature is a table titled 'Pivots' with three columns: 'Name', 'First Regex', and 'Second Regex'. The table contains several rows of pivot definitions:

Name	First Regex	Second Regex
POST_data_services_Manage_	POST /data/services/Manage	
GET_data_Filter1_	GET /data/filter1	
POST_data_services_manageCommercialCustomer...	POST /data/services/manageCommercialCustomers	
POST_data_services_ManageRole_	POST /data/services/ManageRole	
POST_data_services_managelidentAppli_	POST /data/services/managelidentAppli	
GET_data_services_Manage_wsdl_	GET /data/services/Manage?wsdl	

At the bottom of the window, there are five buttons: Parse, Save, SaveAsTemplate, TestRegex, and Cancel.

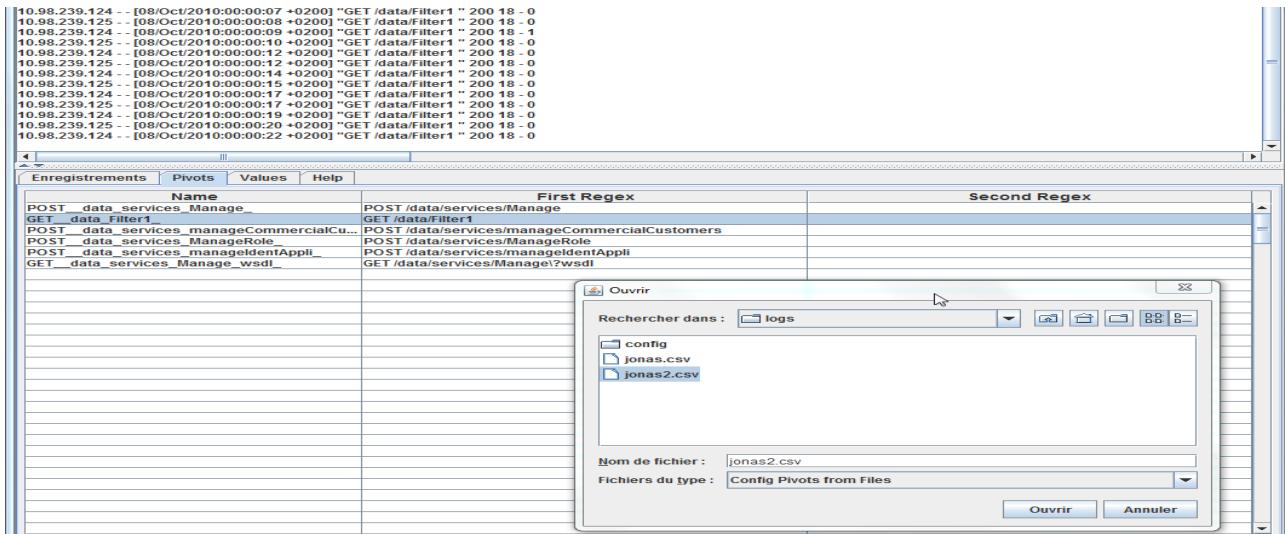
What is a “pivot” ?

In a record, there are other information, that are not numeric, but allow to sort the values. Example URLs in Apache Access Logs, or other WAS access logs , a **Pivot** can be a specific pattern of URL as shown above.

This tab has 3 columns :

- **Name :**
 - The free name of the Pivot
- **First Regex :** You put a full string value if you can or a regexp to extract the value. If you are not able to extract the value with this first regexp, you can apply a second regex (**Second Regex**)
- **Second Regex** to extract the final Pivot from the first matching.

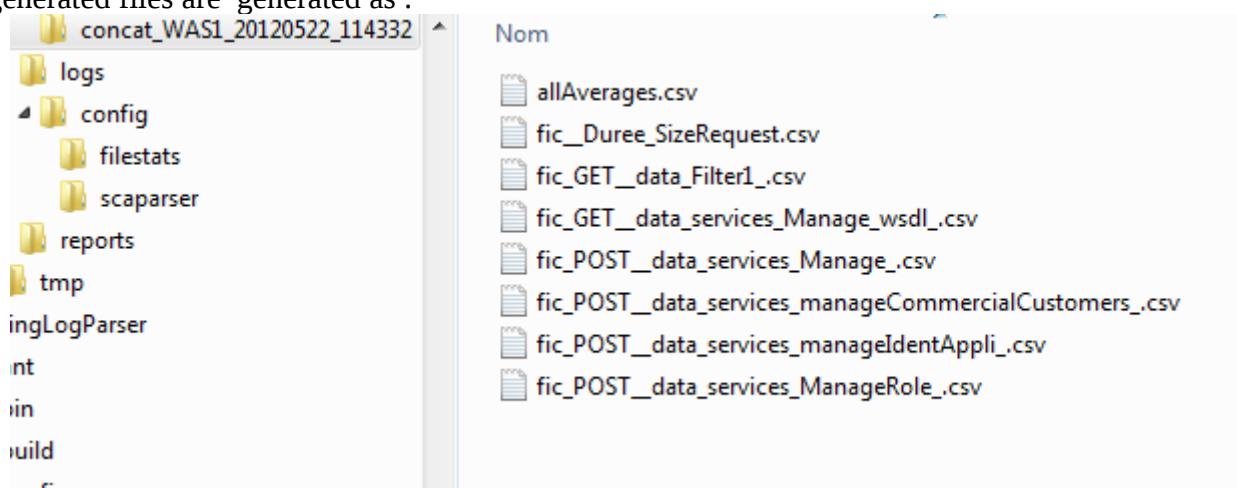
Hint 1 : If beforehand you have parsed this file with **scaFileStats** (see farther in the document) and saved the result table in a csv file in the logs directory, by right clicking in the table Pivot and choosing the csv file saved, the rows of the Pivot tab are automatically filled. After you can suppress, modify, insert other rows.



Hint 2 : For the column **First Regex**, if you use full String as possible (without pattern regex as `\s?\?`...) , the treatment is faster.

Principle of generated files in the csv directory

If you have 2 values in the Tab Value (Duree and SizeRequest) and 6 Pivots in the tab Pivot, the 8 generated files are generated as :



- **fic_Duree_SizeRequest.csv** => global, without Pivot
- The last 6 files are the csv files where each file is bound to a Pivot .
- **allAverages.csv** (Contains the average column of the 7 files above)

Help Tab:

To be filled in a future version. It will describe, like above, how to fill the different tabs and fields.

Buttons :**TestRegex:**

The button **TestRegex** allows to test regex in a Dialog box. See the Menu tool for more explanation.

Save :

The button **Save** saves the configuration and can be reused for parsing another time

SaveAsTemplate :

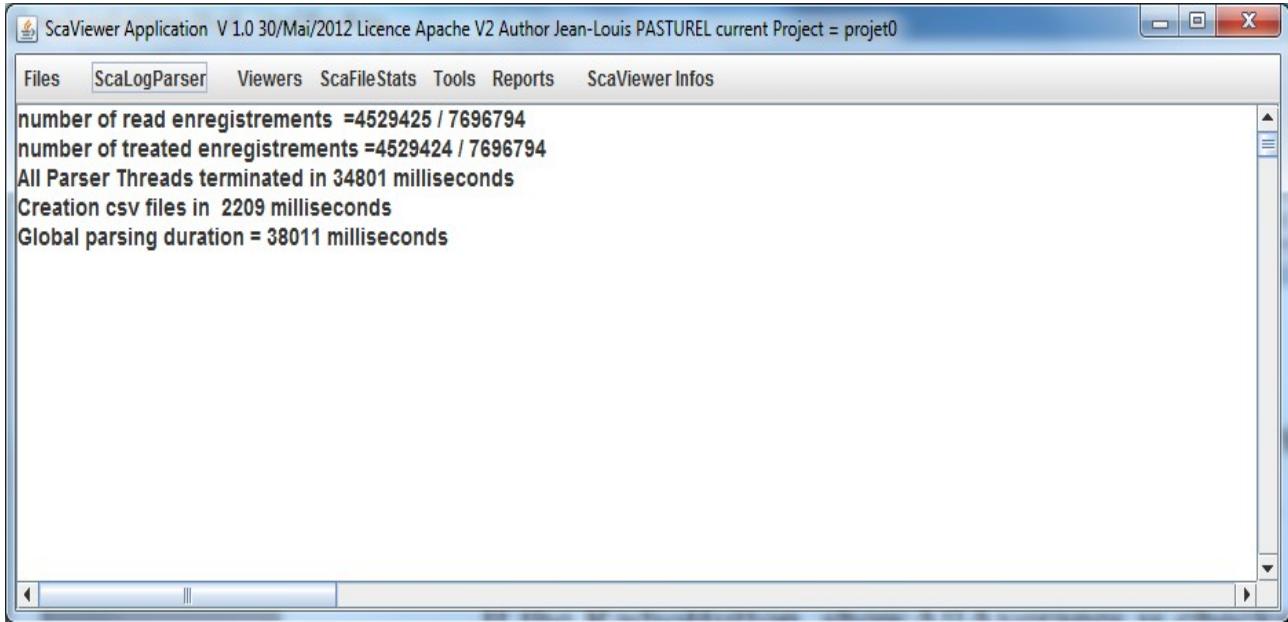
The button **SaveAsTemplate** allows to save a template of the configuration that will be used as pattern for the same kind of access log file.



We can save this template on the context of the project (**RadioButton General Template ?** Unchecked), or in a General Context (can be used by others projects) when **RadioButton General Template ?** is checked.

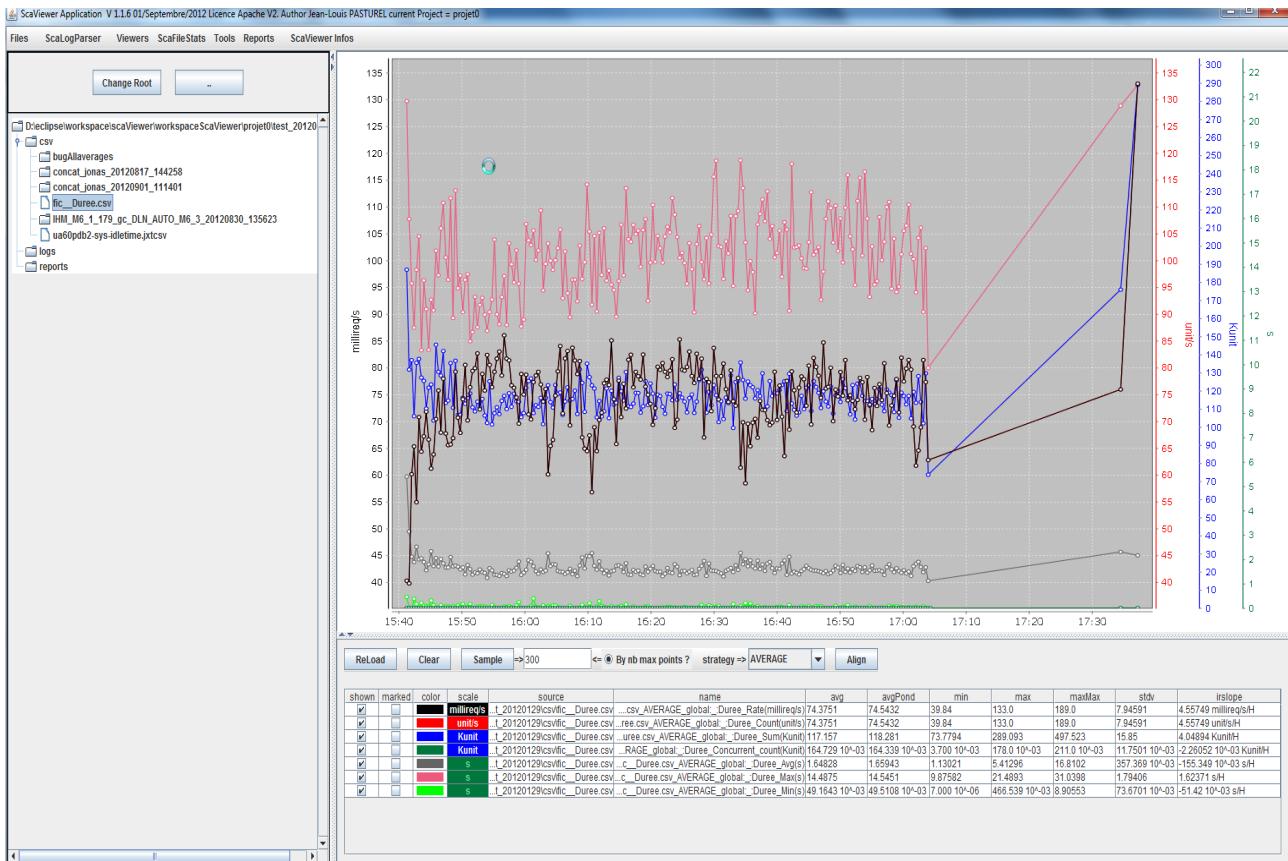
Parse Logs :

The click on Parse Logs starts the Parsing :



This screen is shown at the end of the parsing, without graphing the **allAverages.csv** file because the radioButton **View AllAverages after parsing** was unchecked.

If the RadioButton **View AllAverages after parsing** is checked before launching the parsing, the **allAverages.csv** is graphed as shown below :



3.3 Menu "ScaFileStats"

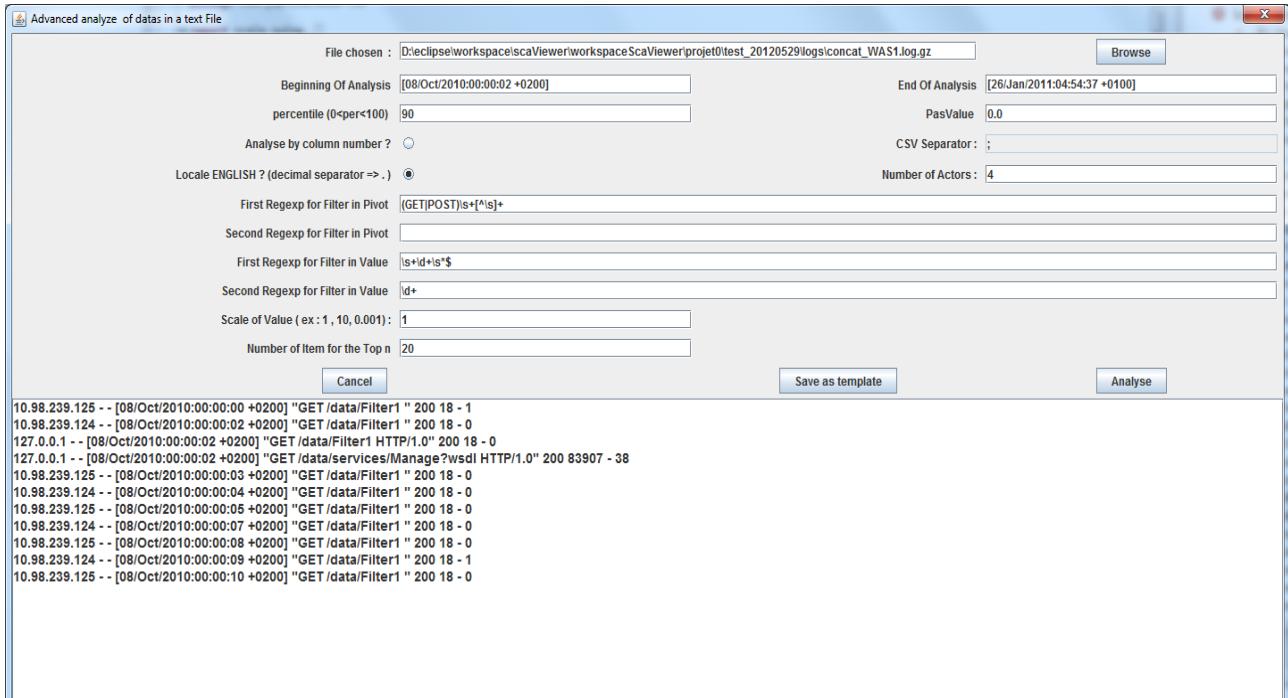
To be used, when the records are mono-line. The feature recognize a certain numbers of date Pattern.

If a pattern is not known, this pattern must be added in the file (advanced topic) :
<swingScaViewer_Home>/config/scaViewerDates.properties



3.3.1 Sous-Menu "StatDatas"

After choosing a log file to parse :



The fields to fill are explained in the table below :

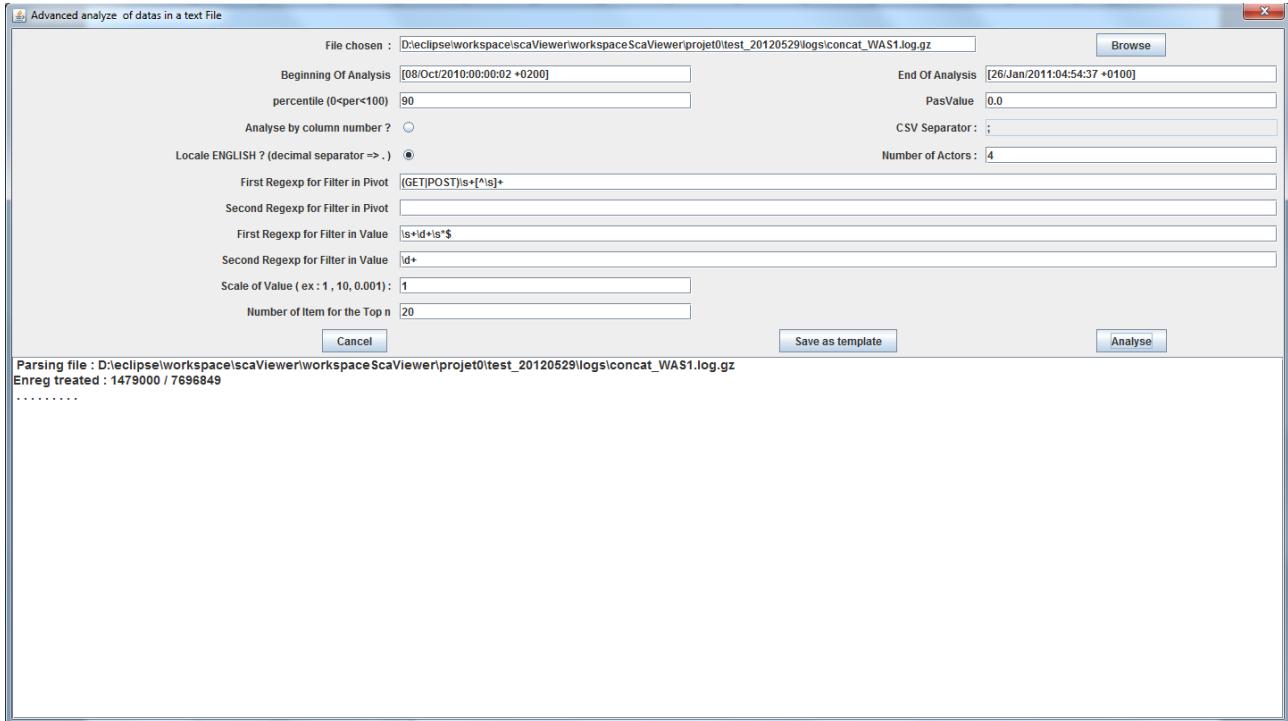
Name of Parameters	Possibles Values	Comments
Beginning Of Analysis	[08/Oct/2010:00: 10:00 +0200]	By default the first date found in the file but can be modified by hand. The format of the date must be respected
End Of Analysis	[26/Jan/2011: 00:30:00 +0100]	By default the last date found in the file but can be modified by hand. The format of the date must be respected
percentile	Number between 0 and 100	The percentile X% : the value which X % of the values are lesser than this value
PasValue	Number depending on scale value	When there is a huge numbers of values in the source file, a OutOfMemory could be raised during the computing of the Mediane Value and the Percentile value. This parameter allows to aggregate values by step, and so the number of values decreases
Analyse by column number	checked/unchecked	If checked, source file is in csv format.
Locale English (decimal separator => .)?	checked/unchecked	If Checked => English => decimal separator = . If not checked => French => decimal separator = ,
Number of Actors	2	Akka Actors to treat the file. =Nb CPUs (if not hyperthreaded) =NbCpus*2 (if hyperthreaded)
Csv separator	, or ; or other character	CSV Separator, enabled when Analyse by column number is checked
First regexp for Filter in pivot	Java/perl regex	First Regexp to extract the Pivot
Second regexp for Filter in pivot	Java/perl regex	If necessary, second regexp to extract the Pivot from the first matching
First regexp for Filter in Value	Java/perl regex	First regexp to extract the value
Second regexp for Filter in Value	Java/perl regex	If necessary, second regexp to extract the Value from the first matching
Scale of a value	1, 10, 0.001 ...	To change the scale of the value
Number of items of the topn	10	Defines the Top n that will be shown after the parsing.

In case of csv file(**Analyse by column number** checked) :

- the field **First regexp for Filter in pivot** is named **index of Column for Pivot (starts to 0)**
- the field **First regexp to extract a Value** is named **index of Column for Value (starts to 0)**

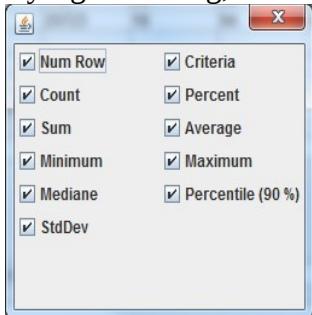
The other fields are unchanged.

After clicking on Analyse button :



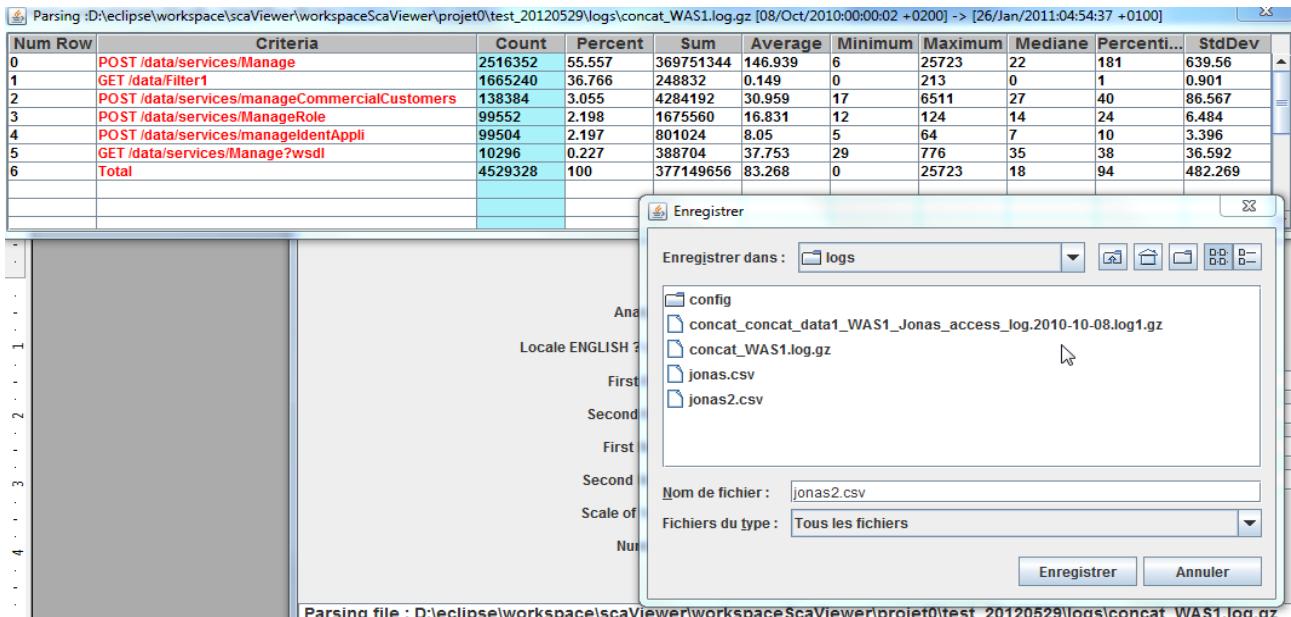
Num Row	Criteria	Count	Percent	Sum	Average	Minimum	Maximum	Mediane	Percentile...	StdDev
0	POST /data/services/Manage	2516352	55.557	369751344	146.939	6	25723	22	181	639.56
1	GET /data/Filter1	1665240	36.766	248832	0.149	0	213	0	1	0.901
2	POST /data/services/manageCommercialCustomers	138384	3.055	4284192	30.959	17	6511	27	40	86.567
3	POST /data/services/ManagerRole	99552	2.198	1675560	16.831	12	124	14	24	6.484
4	POST /data/services/manageIdentAppli	99504	2.197	801024	8.05	5	64	7	10	3.396
5	GET /data/services/Manage?wsdl	10296	0.227	388704	37.753	29	776	35	38	36.592
6	Total	4529328	100	377149656	83.268	0	25723	18	94	482.269

By right clicking, on the title of column (Sum for example), you can remove useless column :



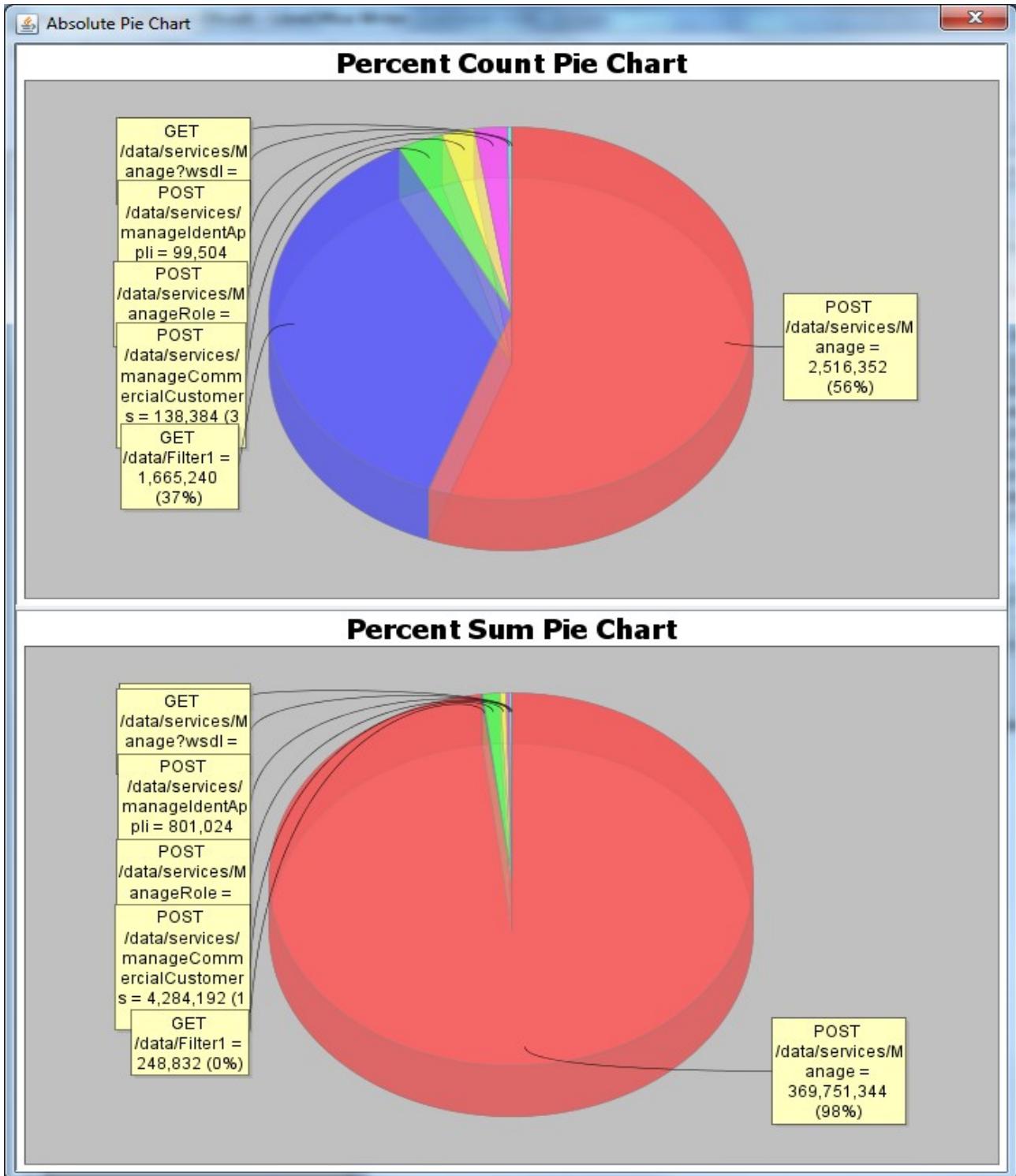
By right clicking, in the contain of the table you can

- save the table in a csv file
- graph the table as Pie Chart.



Hint: Saving this file in Csv Format, allows to fill automatically the Pivot Tab of the Menu ScaLogParser (see Menu ScaLogParser more above in the document)

Example for the Pie Chart :

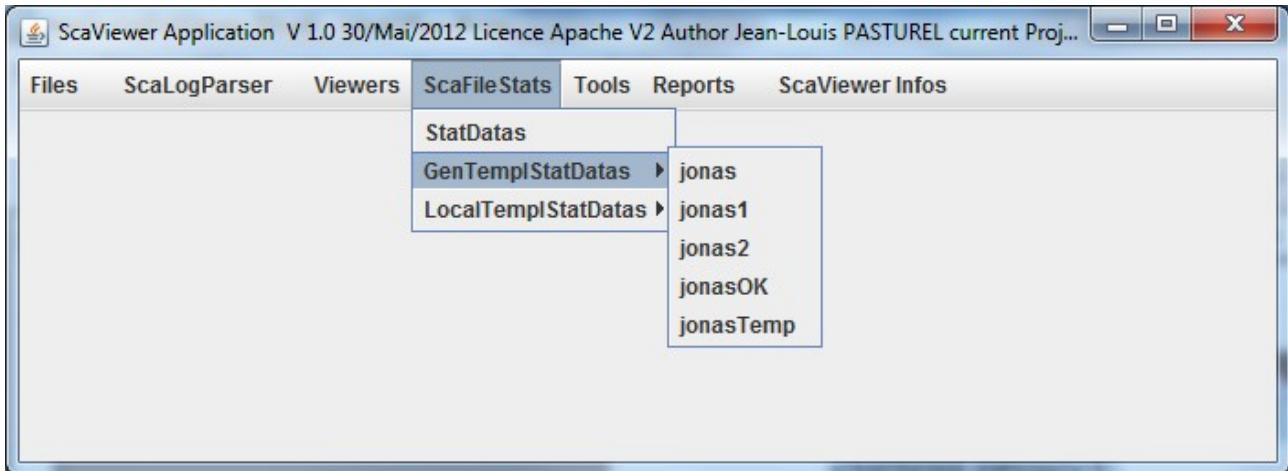


Button SaveAsTemplate

The button **SaveAsTemplate** runs as we have seen above to save a template with **ScaLogParser**. You can save a General Template shared with all projects or a local template used only by the

current project.

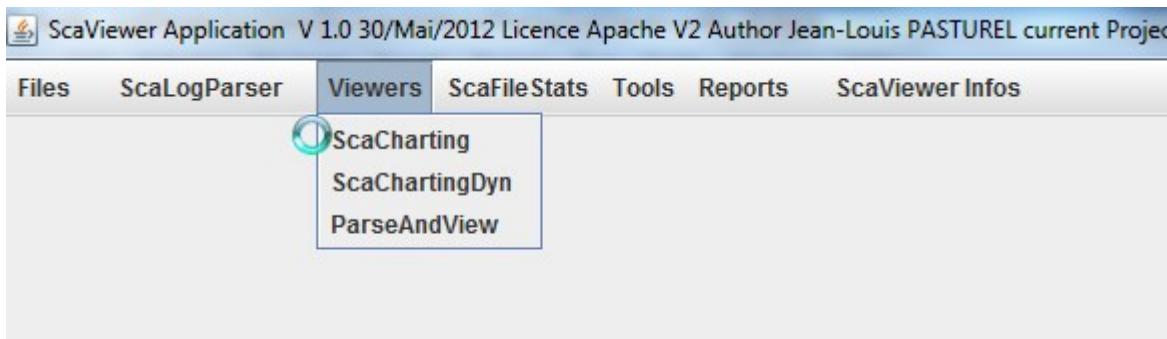
3.3.2 Sub-Menu “GenTemplStatDatas” and “ LocalTemplStatDatas ”



Choose a template before opening a log file.

The fields are filled with the values of the template **after** you have chosen the file to parse

3.4 Menu "Viewers"



Csv file restrictions :

The only CSV files that ScaCharting and ScaChartingDyn can graph must have the following configuration :

- the first line contains title of columns separated by the csv separator
- the first column must be a date respecting one format described in the file
`<swingScaViewer_Home>/config/scaViewerDates.properties`
- The others columns can contains numeric values or strings (Pivots), but a column must be fill with the same type. This column can be empty (if ; is the csv separator ;; indicates an empty column)

This kind of file contains “**Time Series**”, in the framework JFreeChart concepts.

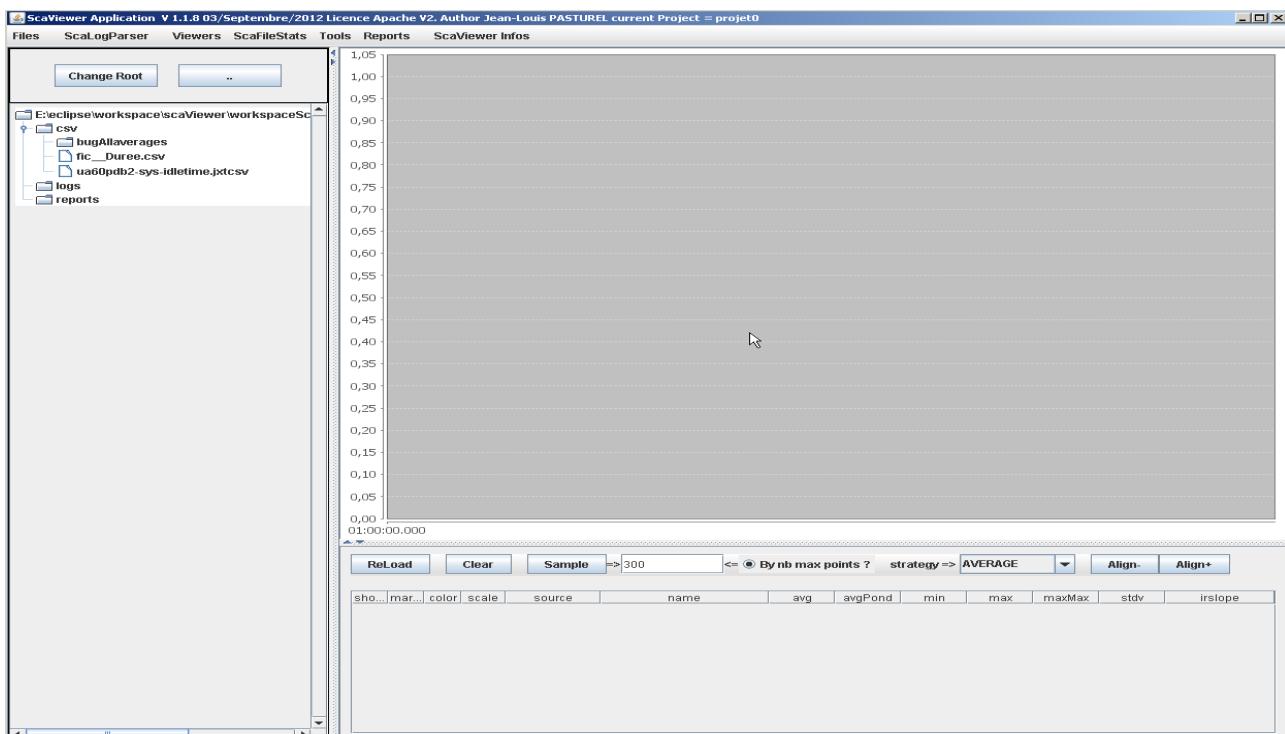
3.4.1 Sub-Menus "ScaCharting" and "ScaChartingDyn"

This 2 sub-menus are similar, the difference is that the ScaChartingDyn sub-menu refreshes periodically the graph if the csv file has changed. The period of refreshing is given by the variable : **scaviewer.dyn.timeout** in the file <swingScaViewer_Home>/config/scaViewer.properties

DnD means Drag and Drop.

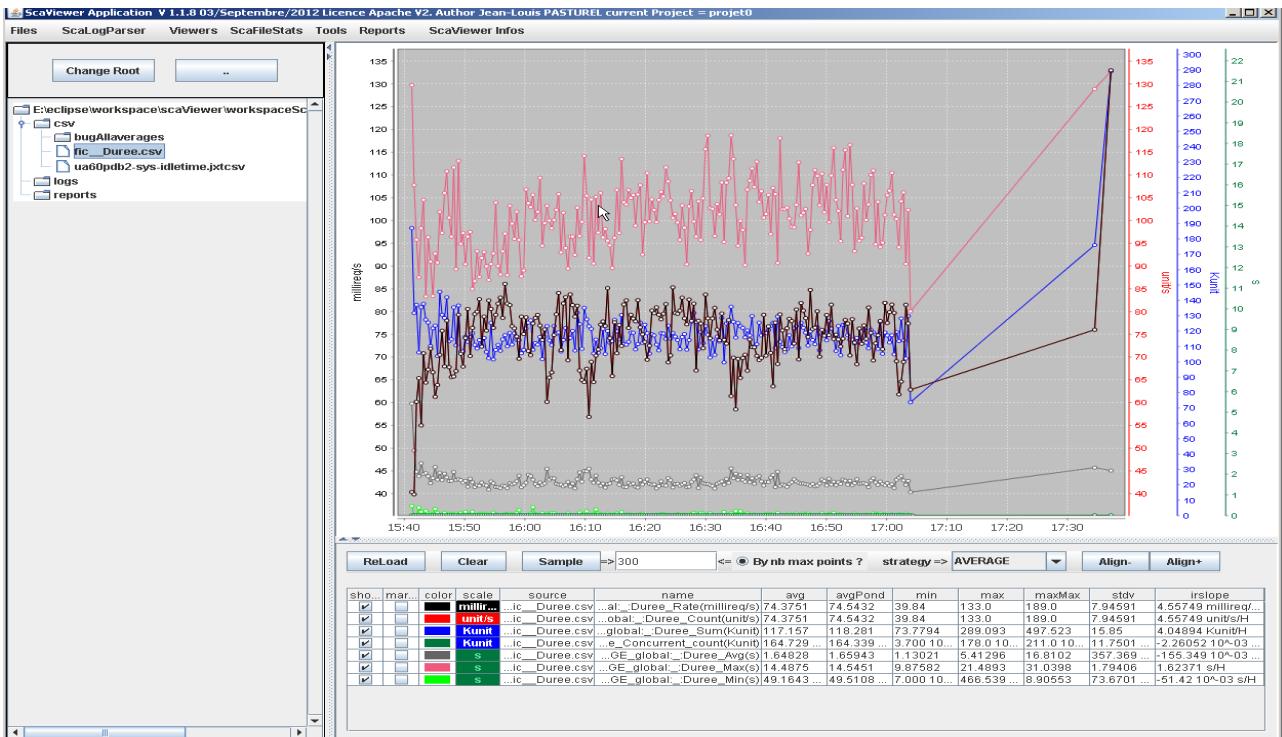
This menu allows to choose csv files, the tree at the left is positioned at the root folder of the current project.

Only the files, where the suffix is in the list of the variable **jtree.suffixes** , are displayed (file <swingScaViewer_Home>/config/scaViewer.properties)



After deploying a branch of a tree by clicking on node, you can drag and drop a csv file or a discontinued list of csv files to the right side of the screen. You can also add csv file by DnD from the same or another directory.

The graph must be zoomed by selecting a region. It can be de-zoomed by dragging on it from the right to the left.



Clicking right on the graph provides also other feature (copy in the clipboard ..)

Clicking right on the title of the table allows to add/remove columns.

Clicking right in the contains of the table allows remove selected row series on the table and on the graph.

Buttons :



Reload

gives the possibility to reload the initial csv files

Clear

Cleanes all CSV file from the Chart and the table

Sample

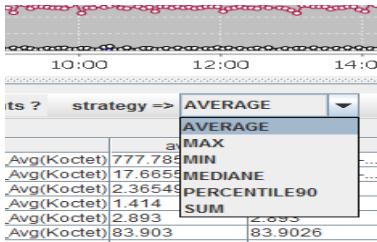
The button is bound with the text field and the radio button at its right

- if the radio button “By nb max points ?” is selected, the chart is shown with the number of points given by the text field (in fact the time interval shown is divided in 300 intervals for the example above)
- if the radio button “By nb max points ?” is unselected, the time interval shown is divided in periods of the given by the text field expressed in milliseconds.

Hint :

When you zoom a part of the chart, click on **Sample** to get more points.

Strategy



When the csv files have more lines than can be displayed on chart (or the number in the text field described just above), you can choose the strategy of aggregation in this combo-box.

Align- : (comes with V 1.1.7)

allows, when 2 or more series are “marked” in the table (column “marked”), to align these series to the left side of the chart by shifting the beginning of all series to the minimum of the beginning of the marked series. It helps for comparison or correlation between series. It can be combined with the “Translate” feature of the table (§ 3.4.3).

Align+ : (comes with V 1.1.8)

allows, when 2 or more series are “marked” in the table (column “marked”), to align these series to the right side of the chart by shifting the beginning of all series to the maximum of the beginning of the marked series. It helps for comparison or correlation between series. It can be combined with the “Translate” feature of the table (§ 3.4.3).

3.4.2 Sub-Menu "ParseAndView"

This sub-menu is for parsing popular logs files that have not to be configured by hands. The principle is to recognise the type of file (JVM GC Logs for examples) and to apply to it a template of scaLogParser.

This templates are located in the directory :

<swingScaViewer_Home>/templates/scaparser/popular

GCHotSpotDS.properties	15/05/2012 18:31	Fichier PROPERTIES	6 Ko
GCHotSpotTS.properties	15/05/2012 18:46	Fichier PROPERTIES	6 Ko
popular.properties	14/05/2012 10:17	Fichier PROPERTIES	1 Ko

The files that you meet here are (but it will be extended in the future) :

The files **GCHotSpotDS.properties** and **GCHotSpotTS.properties** are the templates build from the ScaLogParser menu described more above in this document.

The file **popular.properties** file helps to recognize the “footprint” of the log file. If the file log has no footprint in this file, a popup message is displayed.

popular.properties :

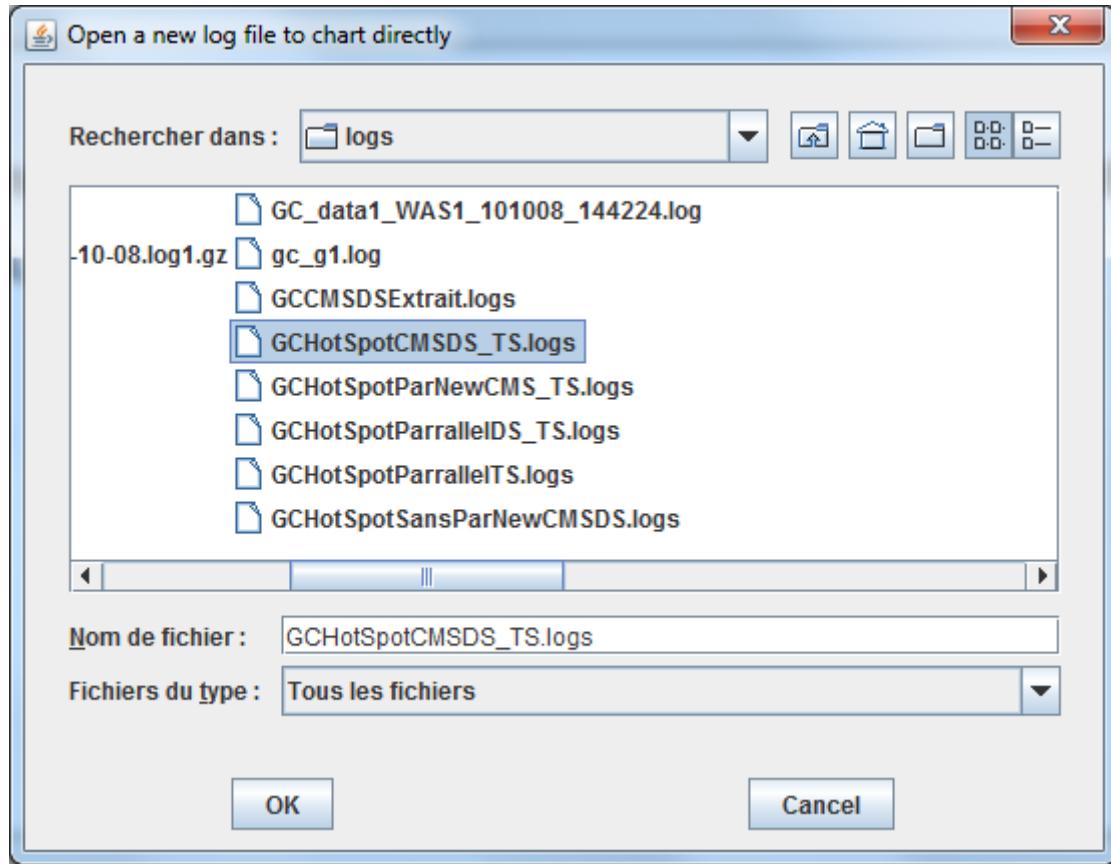
```
popular.list=GCHotSpotDS;GCHotSpotTS

## Empreinte GCHotSpotDS
popular.GCHotSpotDS.debEnr=^(\\d{4}-\\d\\d-\\d\\dT\\d\\d:\\d\\d\\d\\:\\d\\d\\d\\.
\\d\\d\\.\\d\\d\\d(\\+|-)\\d{4}\\:\\s+\\d+\\.\\d+\\:\\s+\\[])
popular.GCHotSpotDS.finEnr=
popular.GCHotSpotDS.isDateExplicit=true
popular.GCHotSpotDS.reg1=(\\[GC|\\[CMS|\\[Full GC)

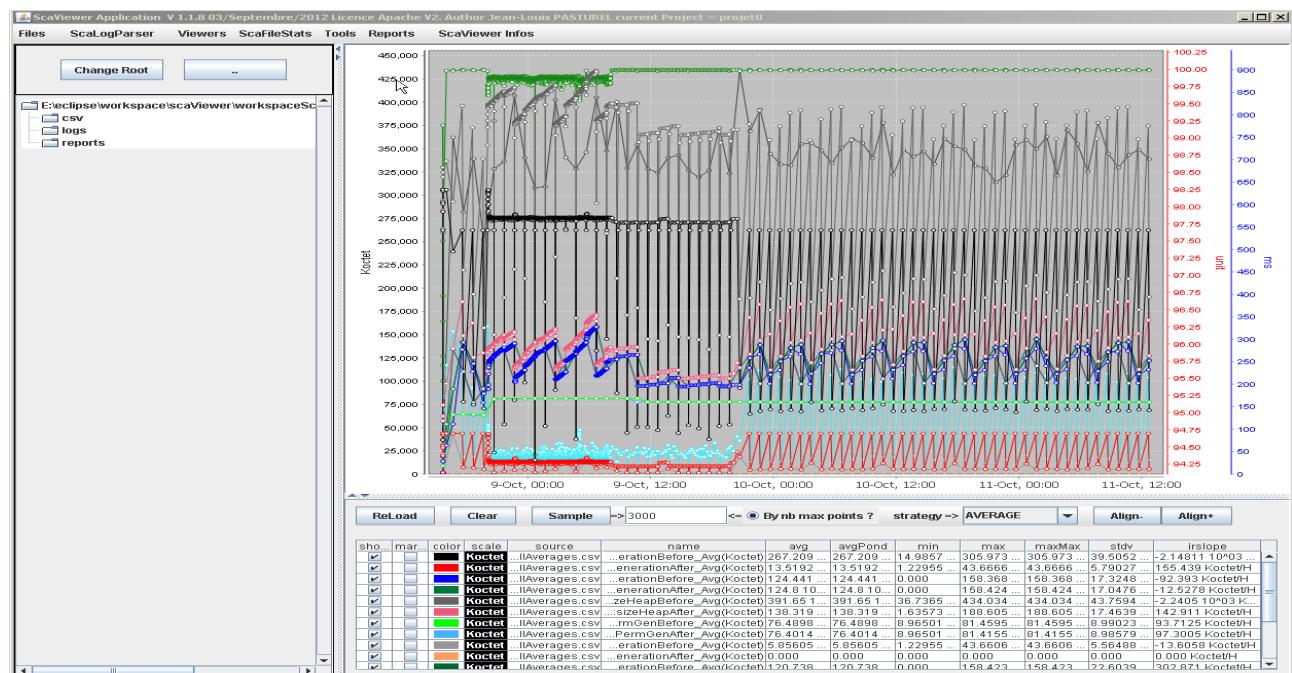
# Empreinte GCHotSpotTS
popular.GCHotSpotTS.debEnr=^(\\d+\\.\\d+:\\s+\\[])
popular.GCHotSpotTS.finEnr=
popular.GCHotSpotTS.isDateExplicit=false
popular.GCHotSpotTS.reg1=(\\[GC|\\[CMS|\\[Full GC)
```

popular.list contains the list of templates treated (without the **.properties** suffix)

The screens are shown below :



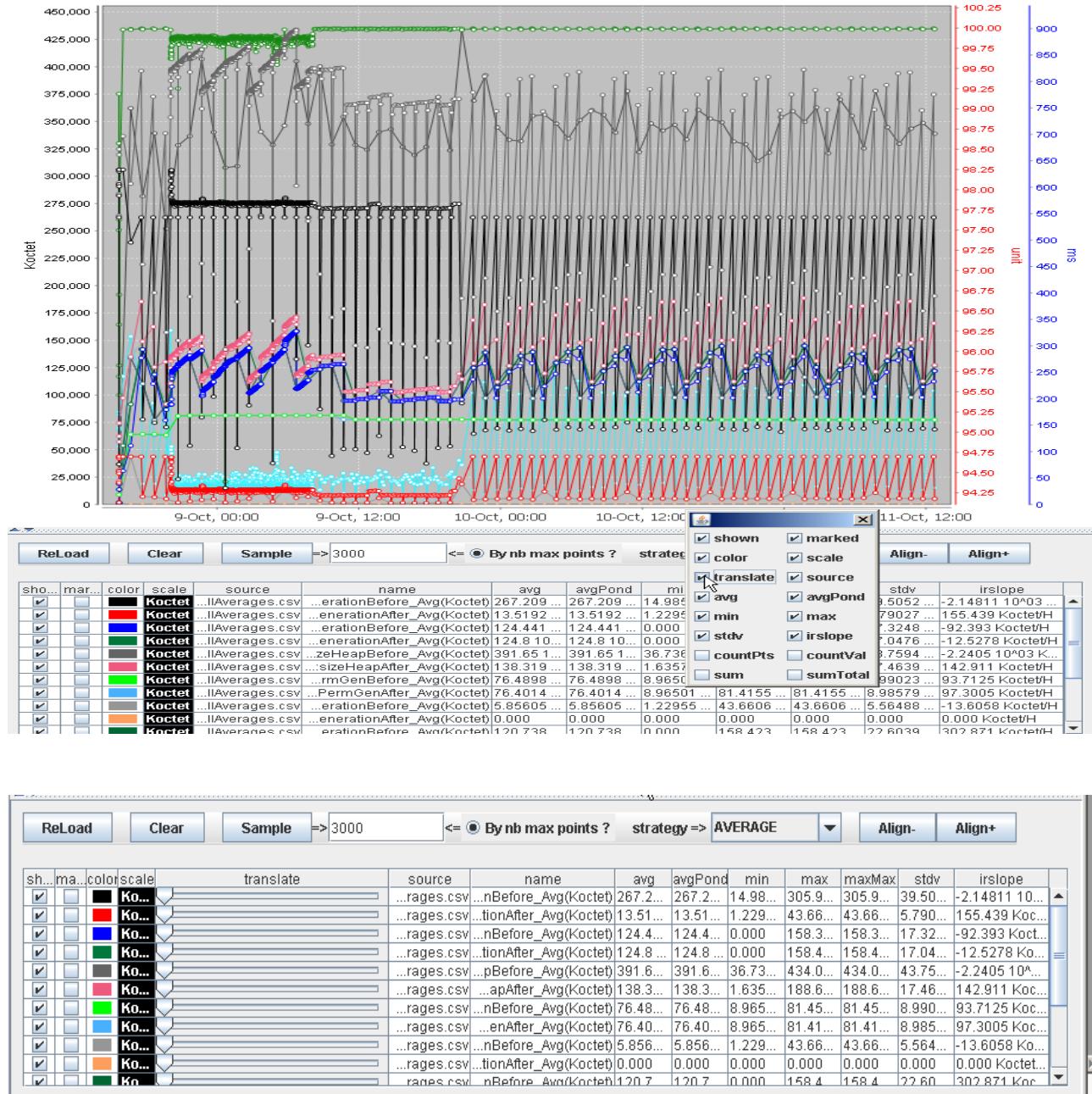
After choosing and click on OK, the chart is directly displayed :



You can go on manipulating the chart as described in the precedent paragraph.

3.4.3 “Funny” feature

When you right click on the title of the table at the button of the Chart window, you can add a column “Translate” :



By playing in the slider you can translate Time Series to the right. It would be useful to compare the behaviour of two time series where the dates are different (every hours, every day, every week ...).

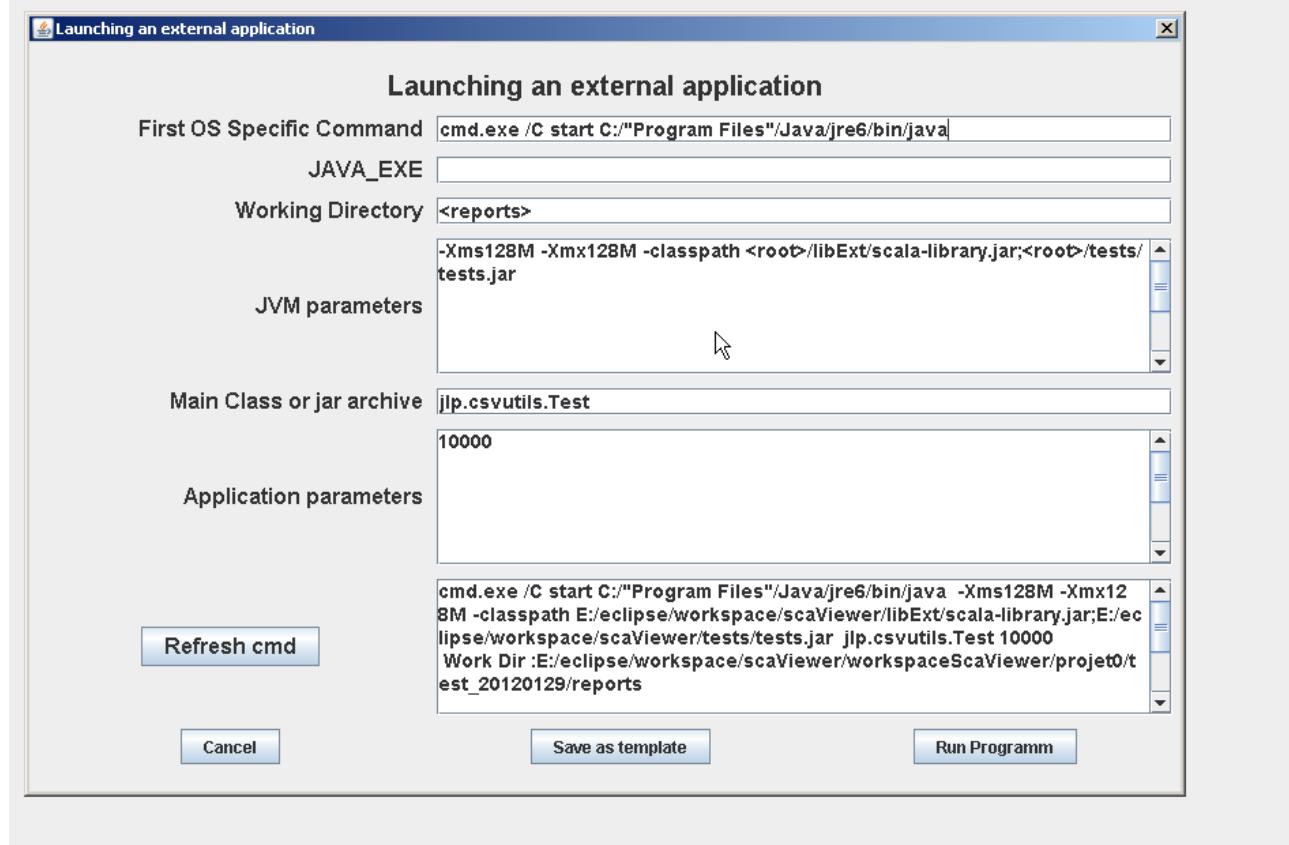
3.5 Menu "MyCommands"

This menu allows to launch external commands, by using results or files located in the swingScaViewer application.

You can create template for interesting external commands. The principle is the same as described for the other precedent menus (ScaFileStats, ScaLogParser)



An example of the main dialog is shown below :



You can launch java application or native command.

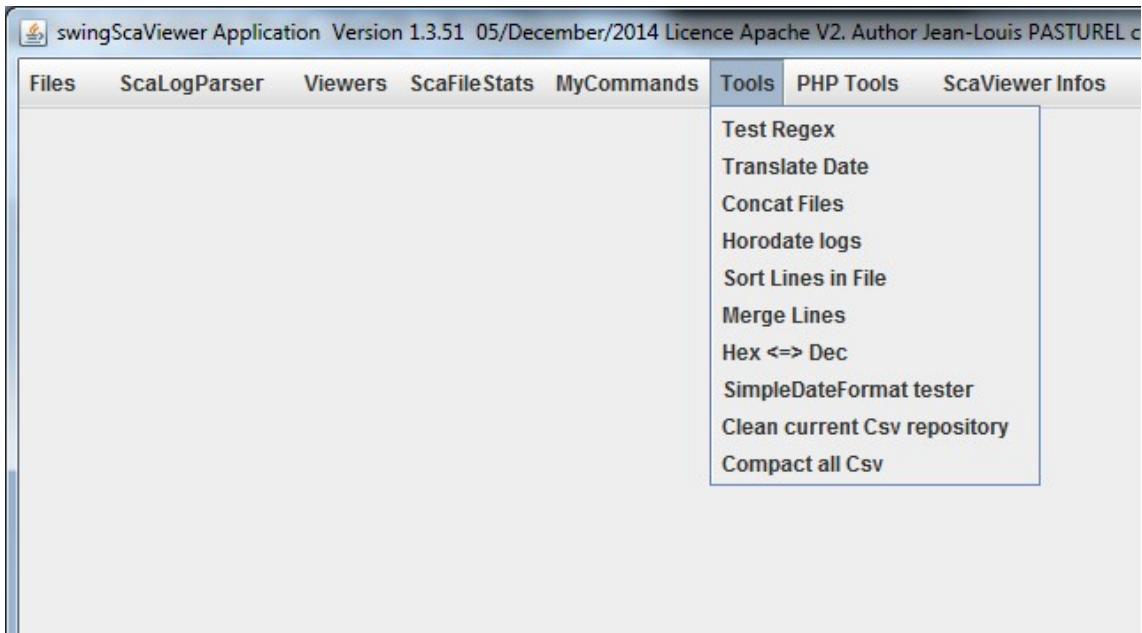
I don't detail the fields, they are explicit.

For each field, you can use patterns ;

- <workspace> the directory of the workspace of swingScaViewer
- <libCommands> the directory jars of the libCommand external commands
- <root> the root directory of the installation of swingScaViewer
- <currentProject> the name of the current project
- <pathYoungTirDir> the last directory created in the current project folder
- <reports> -> the report directory of the last directory created in the current project folder

3.6 Menu "Tools"

This menu groups tools that are useful for swingScaViewer. Other can be added in future versions.



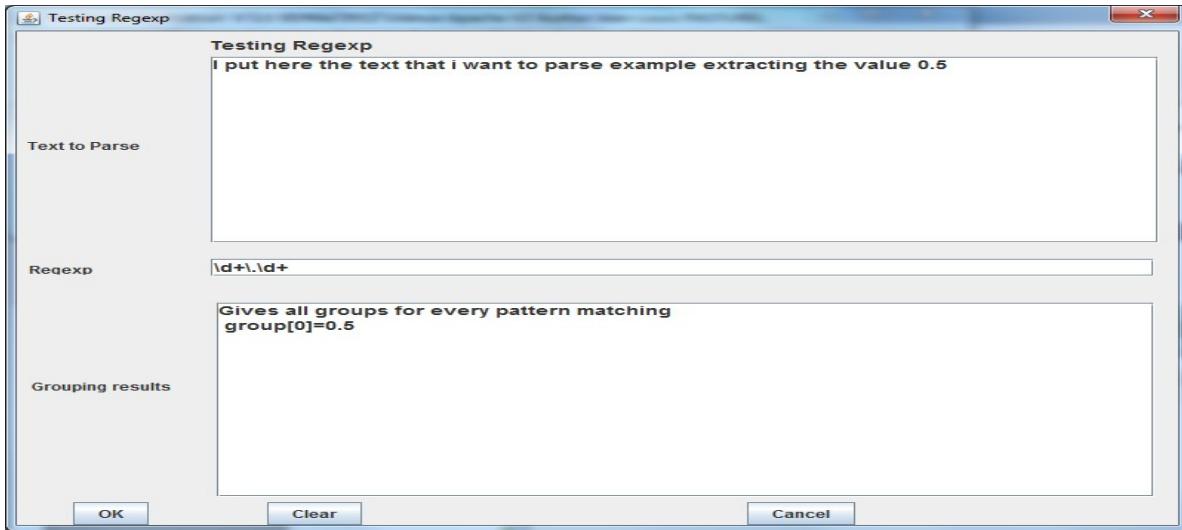
3.6.1 Sub-Menu "TestRegexp"

As seen below, the log parsing uses a lot the mechanism of Regexp Pattern Matching.

The javadoc link is :

<http://java.sun.com/j2se/1.6.0/docs/api/java/util/regex/Pattern.html>

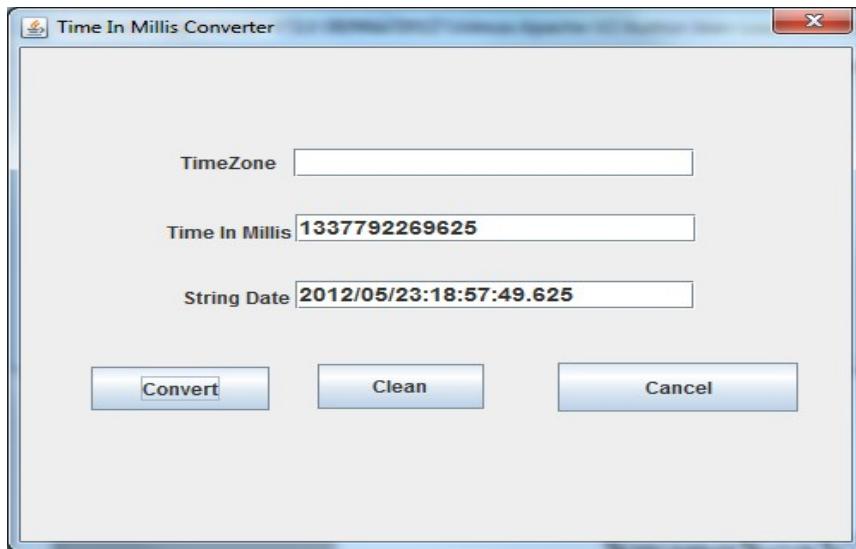
Example :



For swingScaViewer, the interesting result is the result contained in **group[0]**.

3.6.2 Sub-Menu "TranslateDate"

DialogBox that allows date conversions Date in Millis <=> Date in String



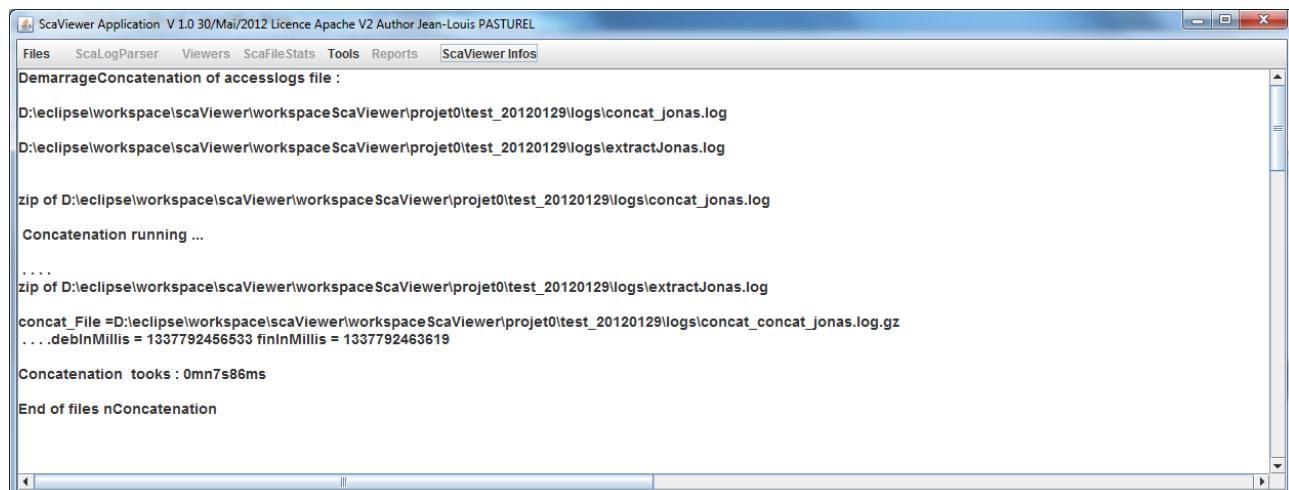
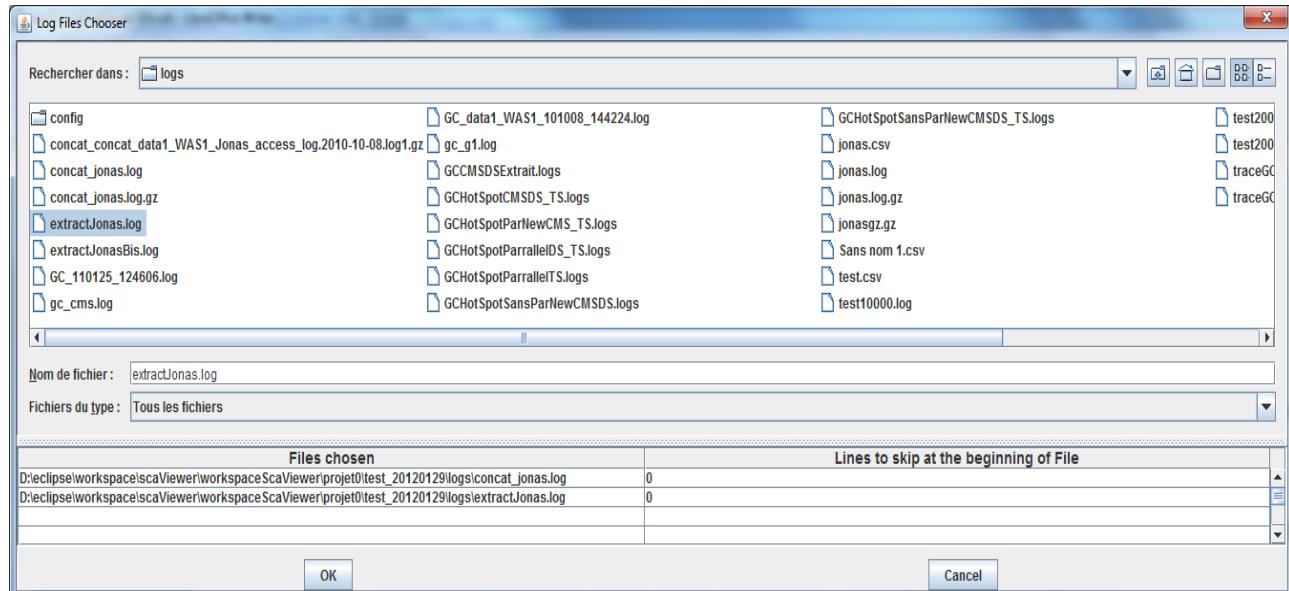
We can :

- not fill any field, and the current date is returned both in milliseconds time and string time
- fill milliseconds Time field and afterwards get the String date
- fill String date field and afterwards get the Milliseconds date
- in the three cases we can modify the Timezone with the format (+|-)dddd => ex : +0200

3.6.3 Sub-Menu "Concat Files"

This sub-menu allows to concatenate files with the same structure in one file. The result file is in gzip format.

The concat files menu, concatenates files, some following the others.



3.6.4 Sub-Menu "Horodate Logs"

It is a tool, that can simplify the parsing of logs files with the structure like below :

```
2012/08/15 10:44:12
useless line1 with a specific pattern
```

```
useless line 2 with a specific pattern
pivot1 value1 value2
pivot2 value1 value2
...
2012/08/15 10:44:14
useless line1 with a specific pattern
useless line 2 with a specific pattern
pivot1 value1 value2
pivot2 value1 value2
...
2012/08/15 10:44:16
useless line1 with a specific pattern
useless line 2 with a specific pattern
pivot1 value1 value2
pivot2 value1 value2
```

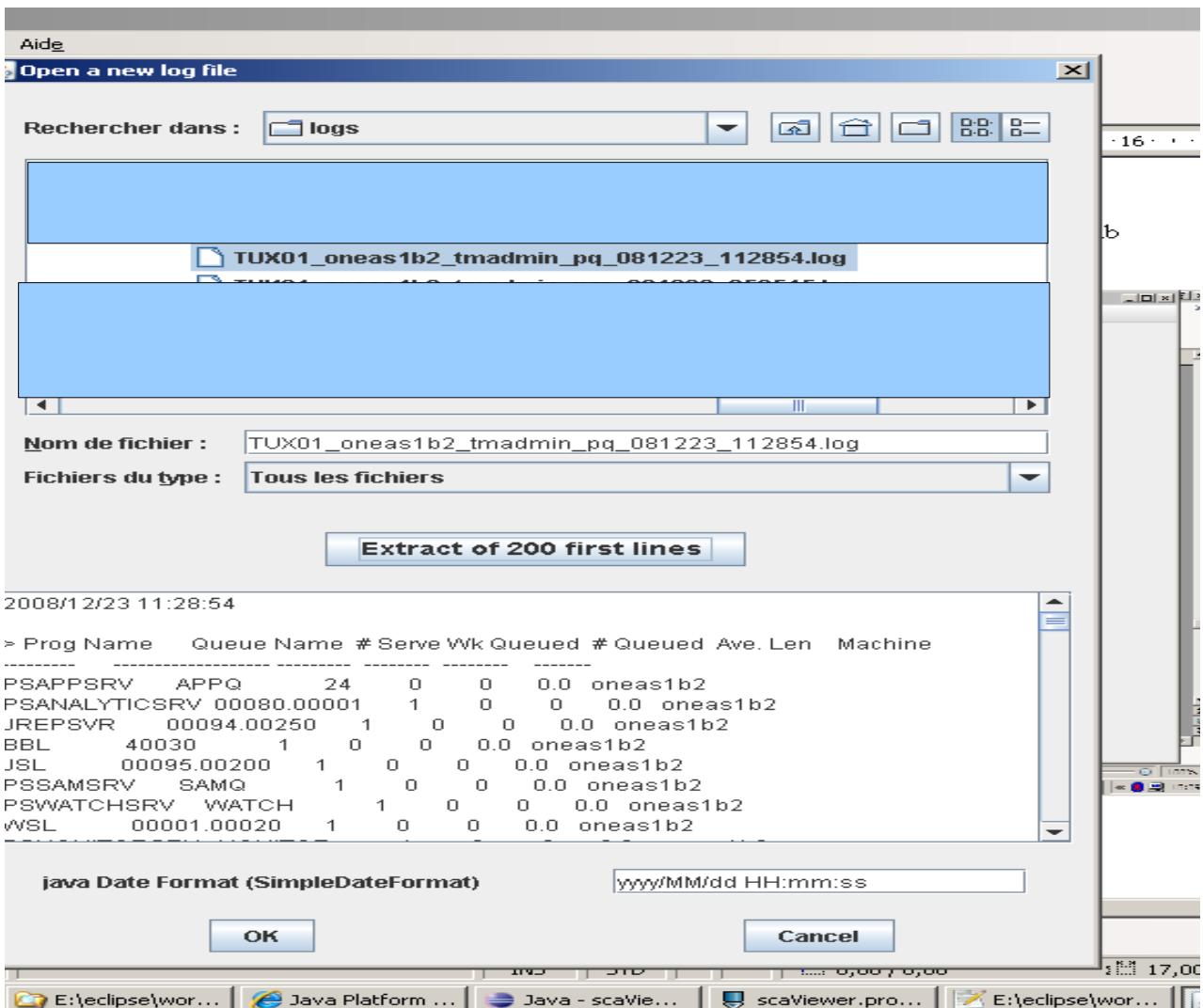
The tool produces a file like this :

```
2012/08/15 10:44:12 useless line1 with a specific pattern
2012/08/15 10:44:12 useless line 2 with a specific pattern
2012/08/15 10:44:12 pivot1 value1 value2
2012/08/15 10:44:12 pivot2 value1 value2
2012/08/15 10:44:12 ...
2012/08/15 10:44:14 useless line1 with a specific pattern
2012/08/15 10:44:14 useless line 2 with a specific pattern
2012/08/15 10:44:14 pivot1 value1 value2
2012/08/15 10:44:14 pivot2 value1 value2
2012/08/15 10:44:14 ...
2012/08/15 10:44:16 useless line1 with a specific pattern
2012/08/15 10:44:16 useless line 2 with a specific pattern
2012/08/15 10:44:16 pivot1 value1 value2
2012/08/15 10:44:16 pivot2 value1 value2
```

This kind of file is easy to parse with scaLogParser. Useless lines can be excluded in the tab “Enregistrements”

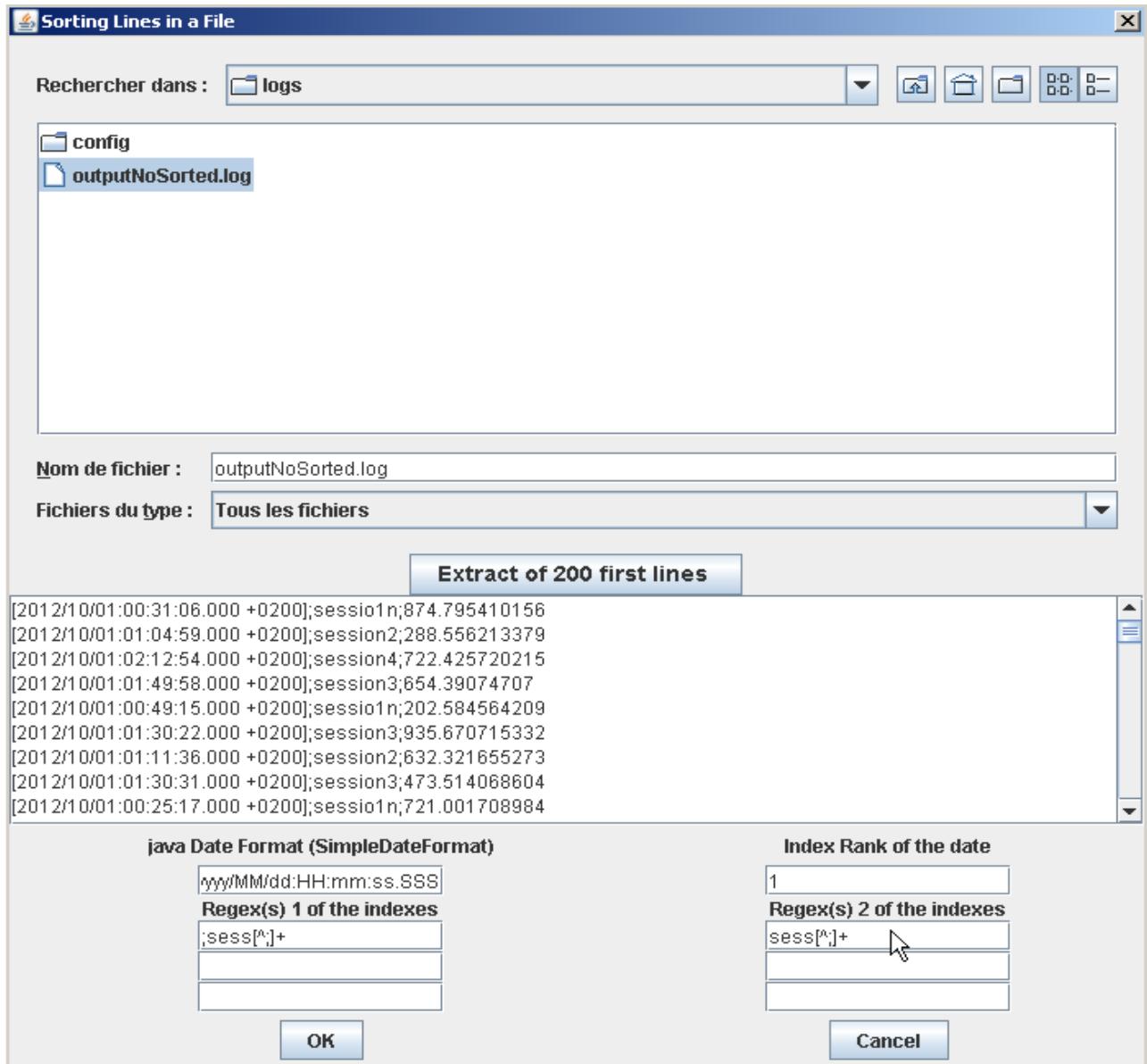
The dialog is shown below.

You have to verify that the java Date Format is correct in regards of the extract of the file. If it isn't, you have to correct it. The java date format must be described in the file
\$root/config/scaViewerDates.properties



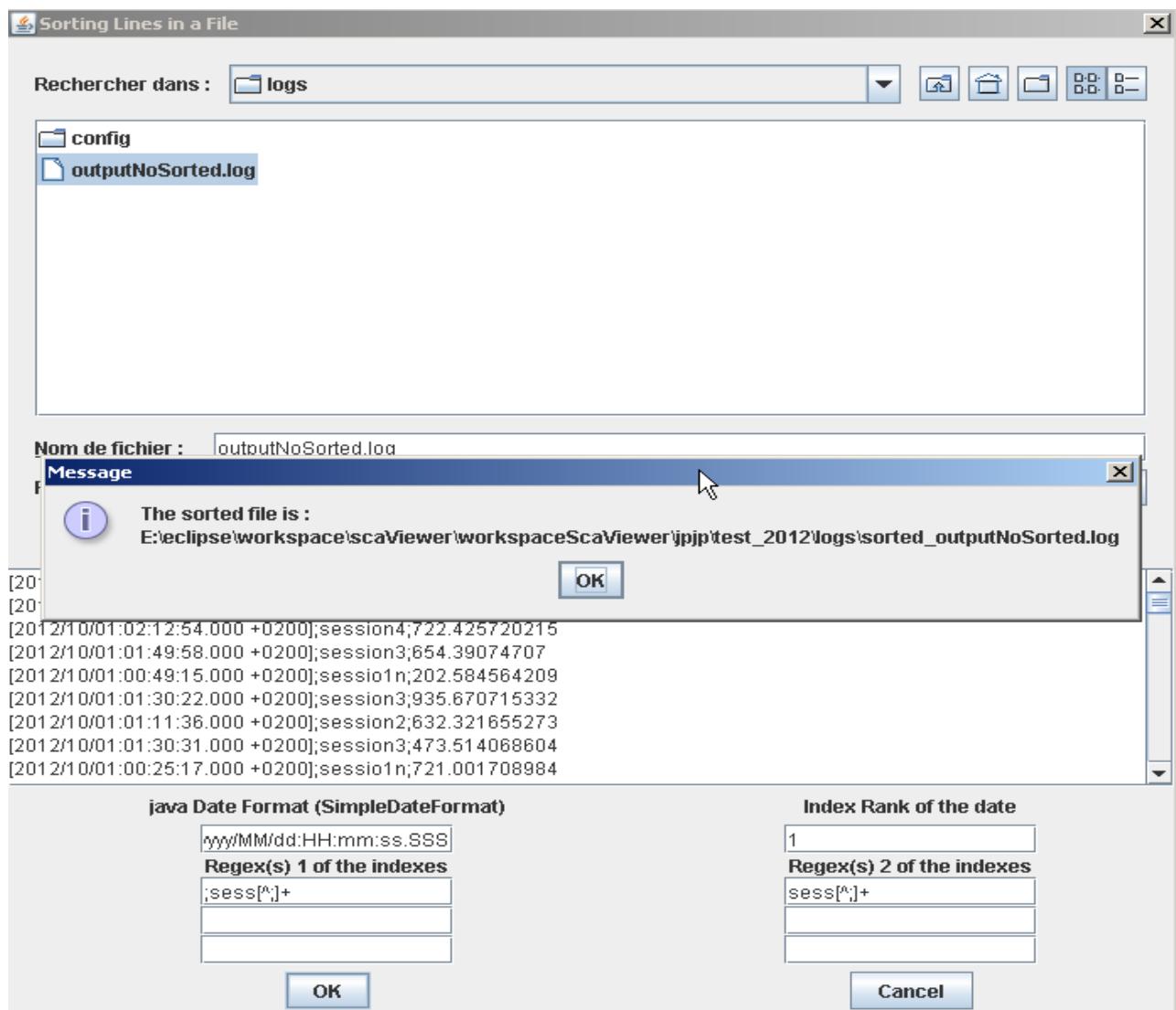
3.6.5 Sub-Menu "Sort Lines In File"

This tool allows, to group lines in a file, that share constant values in this lines. This kind of log files describes different states of a transaction for examples, and the lines are mixed in the file. The mechanist to sort the lines is based on regex, as seen above, with a two pass phases. If the lines contain a date, it can added everywhere in the concatenated index to help to sort the line.



The file contains a pivot sessionId, we want to sort with this pivot, and also with the date at the end; the first index (index 0) is the first production of the first couple of regex, the second index is the date (rank set to 1). If the rank of the date is < 0, the date doesn't participate to the composed index.

After clicking on OK, the sort runs and the result is done on the popup as shown below :



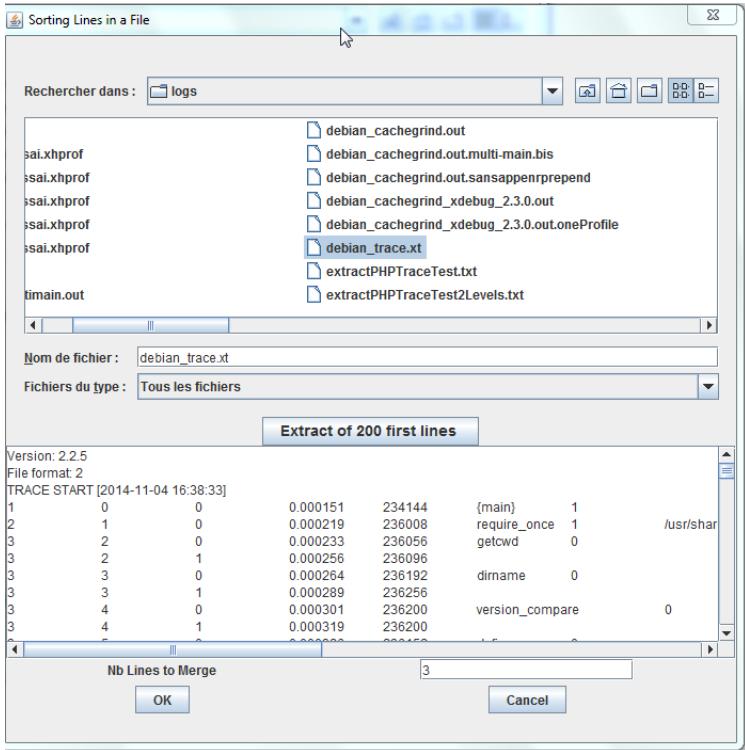
an extract of the new sorted file :

```
1 date;SessionID;valeur
2 [2012/10/01:00:00:00.000 +0200];session1n;896.453674316
3 [2012/10/01:00:00:01.000 +0200];session1n;565.190917969
4 [2012/10/01:00:00:02.000 +0200];session1n;364.403259277
5 [2012/10/01:00:00:03.000 +0200];session1n;527.036804199
6 [2012/10/01:00:00:04.000 +0200];session1n;513.226074219
7 [2012/10/01:00:00:05.000 +0200];session1n;421.101989746
8 [2012/10/01:00:00:06.000 +0200];session1n;170.625595093
9 [2012/10/01:00:00:07.000 +0200];session1n;960.712280273
10 [2012/10/01:00:00:08.000 +0200];session1n;209.536499023
11 [2012/10/01:00:00:09.000 +0200];session1n;383.40625000
12 [2012/10/01:00:00:10.000 +0200];session1n;453.962280273
13 [2012/10/01:00:00:11.000 +0200];session1n;395.857696533
14 [2012/10/01:00:00:12.000 +0200];session1n;365.098632812
15 [2012/10/01:00:00:13.000 +0200];session1n;926.809204102
16 [2012/10/01:00:00:14.000 +0200];session1n;397.192901611
17 [2012/10/01:00:00:15.000 +0200];session1n;623.795166016
18 [2012/10/01:00:00:16.000 +0200];session1n;625.507507324
19 [2012/10/01:00:00:17.000 +0200];session1n;153.096374512
20 [2012/10/01:00:00:18.000 +0200];session1n;408.479034424
21 [2012/10/01:00:00:19.000 +0200];session1n;628.981140137
```

This tool can be useful, when there is a sessionID, with a line for every state of the transaction (Start, running, end). Whith 3 index and the date, we are able to sort the lines, and so the file is parsable with scaLogParser in mutiLine records.

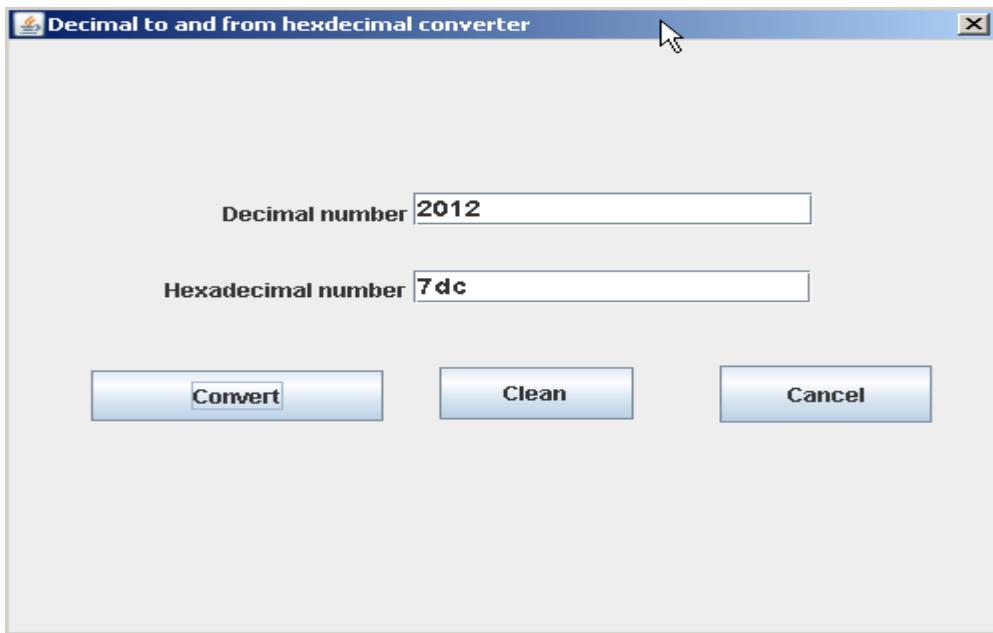
3.6.6 Sub-Menu "Merge Lines"

This tool allows, to group a certain number (Nb lines to Merge) of consecutive lines in a file.



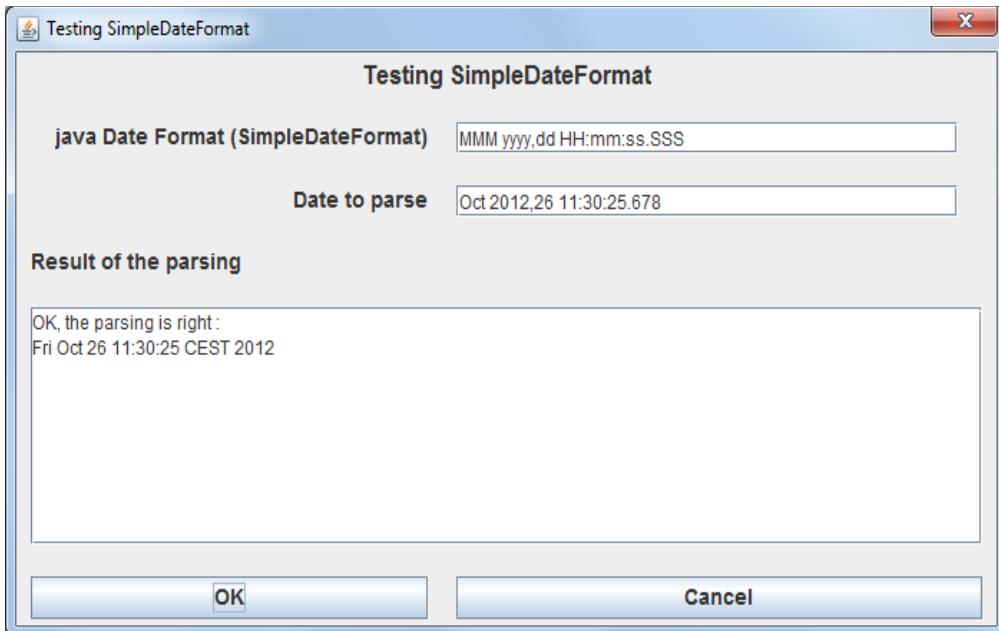
3.6.7 Sub-Menu "Hex <=> Dec"

This tool allow, to convert an hexadecimal number to a decimal number, and the opposite.



3.6.8 Sub-Menu "SimpleDateFormat tester"

This tool allows, to test a SimpleDateFormat against a test date.



3.6.9 Sub-Menu "Clean current csv directory"

Remove all duplicated directories all project csv directory of the youngest scenario (current project/scenario).

The kept directory is the youngest.

A duplicate directory :

- a directory name is compound by <prefix><suffix>
- a suffix is a formatted date as $(-_)?\d\{8\}(-_)\d\{6\}$ corresponding to java SimpleDateFormat : -yyyyMMdd-HHmmss ou _yyyMMdd_HHmmss
- a directory is duplicated if the prefix are the same

3.6.10 Sub-Menu "Compact all Csv"

Remove all duplicated directories (see below for the definition of duplicate directory) in the csv directories for all projects.

The kept directories are the youngest.

Warning : it can take a long time.

3.7 Menu "PHP -TOOLS"

3.7.1 Quick start PHP-TOOLS

Installation of swingScaViewer (Windows)

For Linux not difficulties at all.

- needed a JVM 1.6.0 +
- create a directory <c:\opt>
- unzip in it the archive **swingScaViewer.zip**
- customise the file c:\opt\swingScaViewer\scripts\swingScaViewer.cmd

```
set root=C:\opt\swingScaViewer
set workspace=C:\opt\workspaceLP
```

- create directory c:\opt\workspaceLP
- create a link of c:\opt\swingScaViewer\scripts\swingScaViewer.cmd
- click on it

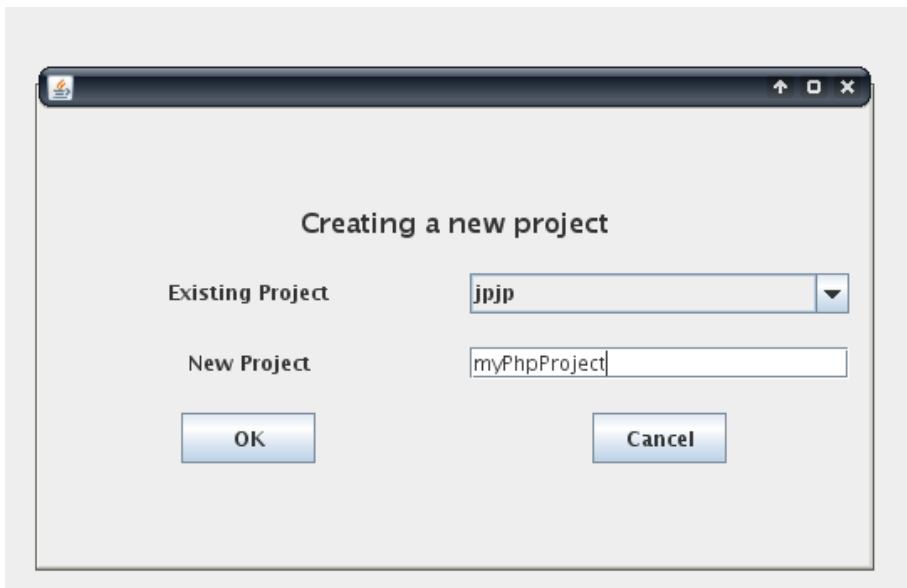
- it runs : waouh !!

- it fails :

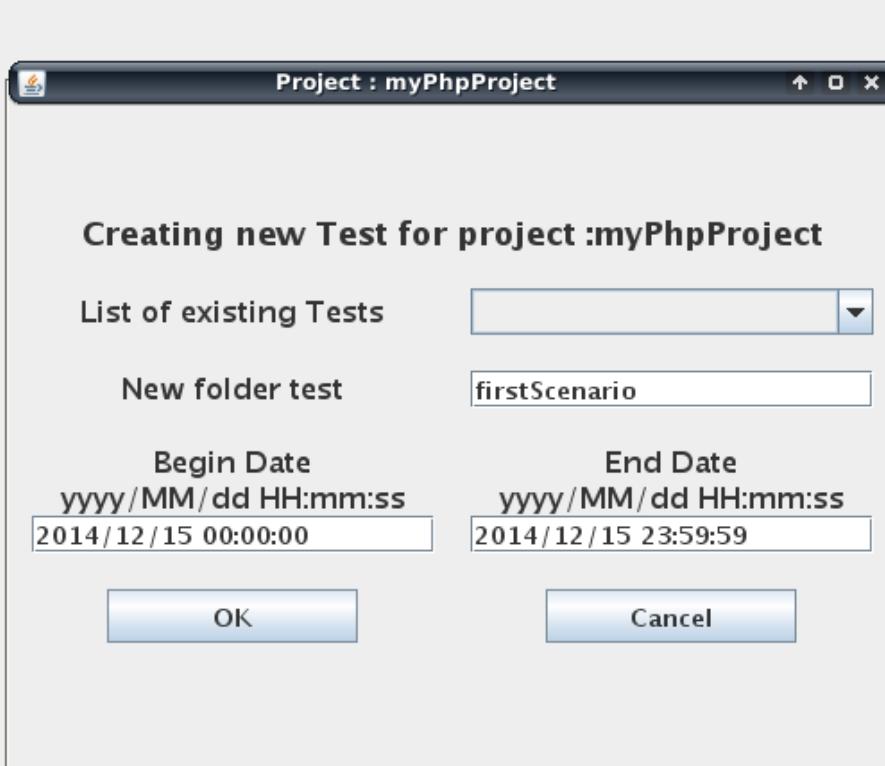
Certainly, it can not find the java executable file. Customize more **swingScaViewer.cmd** with the full path of **java.exe** as below (Don't forget the double quotes) :

```
"C:\Program Files\Java\Jre6\bin\java" %GC_OPTS% -classpath %CLASSPATH%
-Dsun.lang.ClassLoader.allowArraySyntax=true -Dworkspace=%workspace%
-Droot=%root% -Dconfig.file=%root%\config\scaViewer.properties
com.jlp.scaviewer.ui.SwingScaViewer
```

- create a new project (**mandatory**) : Files → New Project



- Create a new scenario (**mandatory**) : Files → Add dir CSV/Logs

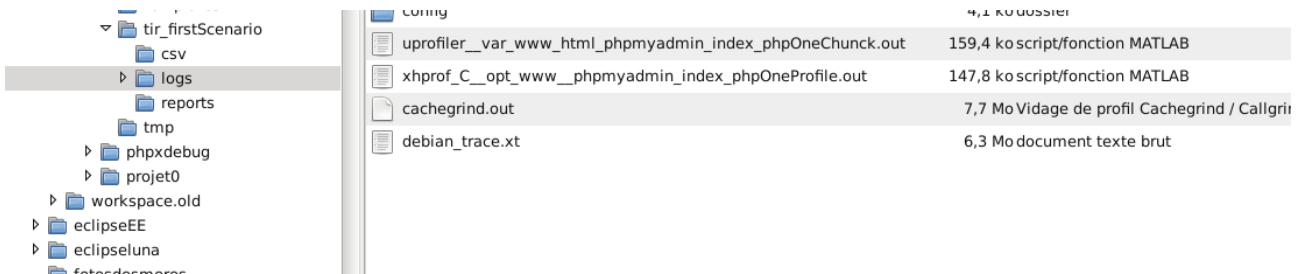


The begin date and end date doesn't matter for PHP Tools for now.

What you have done in c:\opt\workspaceLP:



You have to put the logs in c:\opt\workspaceLP\myPHPProject\tir_firstScenario\logs.
For the demo below, I put these files :



3.7.2 The different ways at stress time

There are 3 ways to profile at stress time with injector tools (Jmeter, Selenium, LoadRunner, Neoload, ...):

Solution 1 :

- instrument **one alone user** in the injection tool (a group of 1 user and a group of n-1 users not instrumented).

Solution 2 :

- randomise the injection of a parameter (XDEBUG_TRACE /XDEBUG_PROFILE for XDEBUG) in the injection script. **Prefer the use of parameter rather than the cookie. The duration of live of the parameter is the duration of the request.** Sometimes only cookie is possible because the application can verify the number of parameters.
- **and** (very important) parameter the name of the output file with pid of process/thread to avoid inter-weaving logs . You can complete the name of the file with the timestamp and the name of the php script.

xdebug.trace_output_name=" %t_%p_wamp_trace_%R"

- to treat with **swingScaViewer** we can concatene all the output files.

Solution 3 :

- set the parameter in a header file (Don't use cookie mechanism for that, if it is possible)
- use the php.ini directive that prepend this header file to all php scripts : **auto-prepend-file**
- as solution 2, to avoid interweaving choose a good name for the output file =>

xdebug.trace_output_name=" %t_%p_wamp_trace_%R"

- to treat with **swingScaViewer** we can concatene all the output files.

Below the third solution is detailed for Xdebug/xhProf/uprofiler .

The archive phptools.zip (or phptools.tar.gz) must be unzipped under the directory DocumentRoot of the application to profile. The files must be available through Apache with an URI like /phptools/*.php

For solution 3, to generate a snapshot of profile with Xdebug/trigger, the method by passing a

parameter (XDEBUG_TRACE / XDEBUG_PROFILE) will be preferred to the method of passing the same with a cookie, **if the application allows it**. It is more simple.

For solution 1 and 2 , by setting parameter/cookie directly by the injection tool, each tool has its own way. I gave more above an example for JMeter. But I think it can be possible with Selenium, Grinder, Gatling, LoadRunner, Neoload ...)

3.7.3 Configuration for Trace format 1/0 and profiling

xdebug.ini

xdebug.ini to be included or in a paragraph of **php.ini**

mode : trace

```
;zend_extension=xdebug.so
zend_extension = "c:/opt/wamp/bin/php/php5.5.12/zend_ext/php_xdebug-2.2.5-5.5-vc11-x86_64.dll"
;
[xdebug]
xdebug.coverage_enable=off

xdebug.remote_enable = off
xdebug.profiler_enable = off
xdebug.profiler_enable_trigger = off
xdebug.profiler_output_name = "%t_%p_debian_%R_cachegrind.out"
xdebug.profiler_output_dir = "c:/opt/tmp"
;xdebug.show_local_vars=0

xdebug.profiler_enable=0
xdebug.profiler_enable_trigger=0
xdebug.profiler_append = 0

xdebug.show_mem_delta=1
xdebug.auto_trace=0
xdebug.trace_enable_trigger=1
xdebug.trace_options = 0
xdebug.collect_params=0
xdebug.collect_return=0
xdebug.trace_format=0
xdebug.trace_output_dir="c:/opt/tmp"
; add %p when more than a user can put the cookie
xdebug.trace_output_name="%t_%p_wamp_trace_%R"

xdebug.file_link_format="xdebug://%f@%l"
;xdebug.remote_log="/var/www/xdebug.log"

; using xdebug with a stressing tool prefer setting a parameter rather than a cookie
; be care to not use the auto_prepend_file, if you inject the QUERY_STRING parameter (XDEBUG_TRACE or XDEBUG_PROFILE) directly from the stress tool
; the purpose of the prepend file is to add this param and redirect ( php function header) the request with this param
; in the auto-prepend_file, there is a control to avoid infinite loop due
; to the automatic add of the auto-prepend_file to all scripts including the redirection by header
auto-prepend_file = "C:/opt/www/phptools/xdebug_header_param_scripts.php"
;auto-prepend_file = "C:/opt/www/phptools/xdebug_trace_header_param_scripts.php"
```

```
; this auto_append_file can fail at first connexion try without it when fails
auto_append_file = "C:/opt/www/phptools/xdebug_footer_cookie_scripts.php"
```

mode : profiler

```
;zend_extension=xdebug.so

zend_extension = "c:/opt/wamp/bin/php/php5.5.12/zend_ext/php_xdebug-2.2.5-5.5-vc11-x86_64.dll"
;
[xdebug]
xdebug.coverage_enable=off

xdebug.remote_enable = off
xdebug.profiler_enable = off
xdebug.profiler_enable_trigger = off
xdebug.profiler_output_name = "%t_%p_debian_%R_cachegrind.out"
xdebug.profiler_output_dir = "c:/opt/tmp"
;xdebug.show_local_vars=0

xdebug.profiler_enable=0
xdebug.profiler_enable_trigger=1
xdebug.profiler_append = 0

xdebug.show_mem_delta=1
xdebug.auto_trace=0
xdebug.trace_enable_trigger=0
xdebug.trace_options = 0
xdebug.collect_params=0
xdebug.collect_return=0
xdebug.trace_format=0
xdebug.trace_output_dir="c:/opt/tmp"
; add %p when more than a user can put the cookie
xdebug.trace_output_name="%t_%p_wamp_trace_%R"

xdebug.file_link_format="xdebug://%f@%l"
;xdebug.remote_log="/var/www/xdebug.log"

; using xdebug with a stressing tool prefer setting a parameter rather than a cookie
; be care to not use the auto_prepend_file, if you inject the QUERY_STRING parameter (XDEBUG_TRACE or XDEBUG_PROFILE) directly from the stress tool
; the purpose of the prepend file is to add this param and redirect ( php function header) the request with this param
; in the auto-prepend_file, there is a control to avoid infinite loop due
; to the automatic add of the auto-prepend_file to all scripts including the redirection by header
auto-prepend_file = "C:/opt/www/phptools/xdebug_header_param_scripts.php"
;auto-prepend_file = "C:/opt/www/phptools/xdebug_header_cookie_scripts.php"
;
; this auto_append_file can fail at first connexion try without it when fails
auto_append_file = "C:/opt/www/phptools/xdebug_footer_cookie_scripts.php"
```

Generally, the only script to include is the prepend file, a footer (for append_file) is also given to kill the cookie (when using the cookie with xdebug) but sometimes it fails (at first connection)

xdebug_header_cookie_scripts.php

```
<?php
if (extension_loaded('xdebug')) {
    include_once 'phptools_incl_commun.php' ;
    if (mustProfile($_SERVER['PHP_SELF']))
    {
        $needCookie=withCookie($_SERVER['PHP_SELF']) ;
        if ( ! $needCookie ){
            $tirage = rand(1,RAND);
            if ($tirage === 1 ) {
                if(ini_get('xdebug.trace_enable_trigger') === '1'){
                    setcookie('XDEBUG_TRACE','1');

                }
                if (ini_get('xdebug.profiler_enable_trigger') === '1'){
                    setcookie('XDEBUG_PROFILE','1');

                }
                register_shutdown_function('xdebug_footer');
            }
        else
        {
            if (isset( $_COOKIE['XDEBUG_TRACE'])){

                setcookie('XDEBUG_TRACE','','',time()-3600);
            }
            if (isset( $_COOKIE['XDEBUG_PROFILE'])){

                setcookie('XDEBUG_PROFILE','','',time()-3600);
            }
        }
    }
}

function xdebug_footer(){

    if (isset( $_COOKIE['XDEBUG_TRACE'])){

        setcookie('XDEBUG_TRACE','','',time()-3600);
    }
    if (isset( $_COOKIE['XDEBUG_PROFILE'])){

        setcookie('XDEBUG_PROFILE','','',time()-3600);
    }
}
```

phptools_incl_commun.php

This file is the same used by the three tools : xdebug/xhprof/uprofiler
In red, the configurations to set (inclusive filter, exclusive filter, randomization).
Read the comments at the beginning of the file for the how-to

```
<?php
// set array of pattern of wanted scripts to profile
// can take 2 forms for value :
//   - withCookie => it is profiled if a cookie XHPROF_PROFILE is present and $tirage === 1
//   - noCookie => always profiled if $tirage === 1

; an array of including filter to be set
define ('PATTERNS',serialize( array(
    '/phpmyadmin/index.php' => 'noCookie',
    "testjlp" => 'noCookie'
    //,'white' => 'noCookie'
```

```
//      , 'version_check.php' => 'noCookie'
 )));

// Some file containing AJAX or JS scripts may not run correctly with xhprof, you can exclude them:
// key is just a mnemonic, the value is the pattern to exclude here
; an array of excluding filter to be set
define ("EXCLUDED" , serialize (array ('javascript' => 'js')));
//define ("EXCLUDED" , serialize (array ()));

// Sampling one script every rand.
// rand = 10 means profile 1 script every 10 runs of the script.
// to always profile a script set RAND to "1"
define ('RAND' , '1');

// to follow the execution of xhprof/uprofiler
// gives a little level of debug
define ('SUIVIPROF', false);

// Truncated : define functions below
...
```

3.7.4 Configuration Xhprof profiler pour swingScaViewer

The files listed below, must be uploaded on the server before the tests. The files ./phptools/*.php must be visible from DocumentRoot of the application with a URI like /phptools/*.php

xhprof.ini

xhprof.ini to be included or a paragraph of **php.ini**
Filters run correctly with xhprof.

```
[xhprof]
;extension= "C:/opt/wamp/bin/php/php5.5.12/ext/php_xhprof_ts.dll"
extension = "xhprof.so"
xhprof.output_dir = "/var/tmp/xhprof"
; For outputs to be parsed by swingScaViewer use these scripts :
auto_prepend_file = "/var/www/html/phptools/xhprof_header_scripts.php"

; For outputs to be parsed by native tools coming with xhprof, use these scripts :
;auto_prepend_file = "/var/www/html/phptools/xhprof_header_natif.php"

;
; to get the 2 files use this prepend-file
;auto_prepend_file = "/var/www/html/phptools/xhprof_header_mixed.php"
```

xhprof_header_scripts.php

```
<?php
if (extension_loaded('xhprof')) {
// see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;
    if (mustProfile($_SERVER['PHP_SELF']))
        {
            $needCookie=withCookie($_SERVER['PHP_SELF']) ;
            if ( ! $needCookie || (($needCookie) && (isset($_COOKIE['XHPROF_PROFILE'])))){
                // set the RAND variable in file xhprof_incl_commun.php to the frequency of
profiling that you want; 10 means : 1 profiled script each 10 runs.

                $tirage = rand(1,RAND);
                if ($tirage === 1 ) {
                    if( SUIVIPROF === true )
                        {
                            $t = microtime(true);
                            $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                            $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                            $date_format=$date->format ( "Y-m-d:H:i:s.u" );

                            $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
                            $treat=$date_format.";Begin treatment of  ".
$_SERVER['PHP_SELF'].";\n";
                            file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                        }
                    register_shutdown_function('footer_scripts');
                    xhprof_enable(XHPROF_FLAGS_CPU | XHPROF_FLAGS_MEMORY);
                }
            }
        }
}
```

```

        }

function footer_scripts(){
    $xhprof_data=xhprof_disable();
    $datas=print_r($xhprof_data, true);
    $id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", " ", $_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
    $t = microtime(true);
    $micro = sprintf("%06d", ($t - floor($t)) * 1000000);
    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
    $date_format=$date->format ( "Y-m-d:H:i:s.u" );
    $firstLine = "## NEW PROFILE FILE BEGIN\n";
    $lastLine="## NEW PROFILE FILE END=".$date_format."\n";
    if( SUIVIPROF === true ) {
        $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
        $treat=$date_format.";End treatment of ". $_SERVER['PHP_SELF'].";\n";
        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
    }
    $toWrite=$firstLine."cmd=".$_SERVER['DOCUMENT_ROOT']. $_SERVER['PHP_SELF']."\n".
$datas."\n".$lastLine."\n";
    $nameFile=ini_get('xhprof.output_dir')."/xhprof_.$id.".out";
    file_put_contents($nameFile, $toWrite, FILE_APPEND | LOCK_EX );
    if (isset( $_COOKIE["XHPROF_PROFILE"])) {
        setcookie("XHPROF_PROFILE", "", time()-3600);
    }
}

```

phptools_incl_commun.php

This file is the same used by the three tools : xdebug/xhprof/uprofiler

In red, the configurations to set (inclusive filter, exclusive filter, randomization).

Read the comments at the beginning of the file for the how-to

```

<?php
// set array of pattern of wanted scripts to profile
    // can take 2 forms for value :
    //   - withCookie => it is profiled if a cookie XHPROF_PROFILE is present and $tirage === 1
    //   - noCookie => always profiled if $tirage === 1

; an array of including filter to be set
define ('PATTERNS',serialize( array(
    '/phpmyadmin/index.php' => 'noCookie',
    'testjlp' => 'noCookie'
//,'white' => 'noCookie'
//    , 'version_check.php' => 'noCookie'
    )));

// Some file containing AJAX or JS scripts may not run correctly with xhprof, you can exclude them:
// key is just a mnemonic, the value is the pattern to exclude here
; an array of excluding filter to be set
define ("EXCLUDED" , serialize (array ('javascript' => 'js')));
//define ("EXCLUDED" , serialize (array ()));

// Sampling one script every rand.
// rand = 10 means profile 1 script every 10 runs of the script.
// to always profile a script set RAND to "1"
define ('RAND' , '1');

// to follow the execution of xhprof/uprofiler
// gives a little level of debug
define ('SUIVIPROF', false);

```

```
// truncated : defining functions below
...
```

xhprof_header_natif.php

```
<?php
if (extension_loaded('xhprof')) {
// see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;
    if (mustProfile($_SERVER['PHP_SELF'])) {
        {
            $needCookie=withCookie($_SERVER['PHP_SELF']) ;
            if ( ! $needCookie || (($needCookie) && (isset($_COOKIE['XHPROF_PROFILE'])))){
                // set the RAND variable in file xhprof_incl_commun.php to the frequency of
                profiling that you want; 10 means : 1 profiled script each 10 runs.

                $tirage = rand(1,RAND);
                if ($tirage === 1 ) {
                    if( SUIVIPROF === true ) {
                        $t = microtime(true);
                        $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                        $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                        $date_format=$date->format ( "Y-m-d:H:i:s.u" );

                        $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
                        $treat=$date_format.";Begin treatment of  ".
$_SERVER['PHP_SELF'].";\n";
                        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                    }
                    register_shutdown_function('footer_natif');
                    xhprof_enable(XHPROF_FLAGS_CPU | XHPROF_FLAGS_MEMORY);
                }
            }
        }
    }
}

function footer_natif(){
    $xhprof_data=xhprof_disable();
    $datas=print_r($xhprof_data, true);
    // The script is really profiled
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/xhprof_lib/utils/xhprof_lib.php";
    include_once $_SERVER['DOCUMENT_ROOT'] .
"/phptools/xhprof_lib/utils/xhprof_runs.php";
    $xhprof_runs = new XHProfRuns_Default();
    if( SUIVIPROF === true ) {
        $t = microtime(true);
        $micro = sprintf("%06d",($t - floor($t)) * 1000000);
        $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
        $date_format=$date->format ( "Y-m-d:H:i:s.u" );
        $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
        $treat=$date_format.";End treatment of  ". $_SERVER['PHP_SELF'].";\n";
        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
    }
    $id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", " _ " ,
$_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
    $nameFile=ini_get('xhprof.output_dir')."/xhprof_.$id.".out";
    $run_id = $xhprof_runs->save_run($xhprof_data,$id);
    if (isset( $_COOKIE['XHPROF_PROFILE'] ) ) {
```

```
        setcookie("XHPROF_PROFILE", "", time() - 3600);
    }
}
```

xhprof_header_mixed.php

```
<?php
if (extension_loaded('xhprof')) {
// see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;
    if (mustProfile($_SERVER['PHP_SELF']))
    {
        $needCookie=withCookie($_SERVER['PHP_SELF']) ;
        if ( ! $needCookie || (($needCookie) && (isset($_COOKIE['XHPROF_PROFILE'])))){
            // set the RAND variable in file xhprof_incl_commun.php to the frequency of
profiling that you want; 10 means : 1 profiled script each 10 runs.

            $tirage = rand(1,RAND);
            if ($tirage === 1 ) {
                if( SUTVIPROF === true )      {
                    $t = microtime(true);
                    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                    $date_format=$date->format ( "Y-m-d:H:i:s.u" );

                    $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
                    $treat=$date_format.";Begin treatment of ".
$_SERVER['PHP_SELF'].";\n";
                    file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                }
            }
        }
    }
}

function footer_mixed(){
    $xhprof_data=xhprof_disable();
    $datas=print_r($xhprof_data, true);
    // Generation file for native xhprof visualisation tools
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/xhprof_lib/utils/xhprof_lib.php";
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/xhprof_lib/utils/xhprof_runs.php";
    $xhprof_runs = new XHProfRuns_Default();
    $id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", " ", $_SERVER['DOCUMENT_ROOT'] . $_SERVER['PHP_SELF'])));
    $run_id = $xhprof_runs->save_run($xhprof_data,$id);
    // Generation file for swingScaViewer tool
    $nameFile=ini_get('xhprof.output_dir')."/xhprof_.$id.".out";
    $t = microtime(true);
    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
    $date_format=$date->format ( "Y-m-d:H:i:s.u" );
    $firstLine = "## NEW PROFILE FILE BEGIN\n";
    $lastLine="## NEW PROFILE FILE END=".$date_format."\n";
    if( $suiviprof === true )  {
        $nameFileSuivi=ini_get('xhprof.output_dir')."/xhprof_suivi.out";
        $treat=$date_format.";End treatment of ". $_SERVER['PHP_SELF'].";\n";
        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
    }
    $toWrite=$firstLine."cmd=".$_SERVER['DOCUMENT_ROOT'].$_SERVER['PHP_SELF']."\n".$datas."\n".
$lastLine."\n";
    $nameFile=ini_get('xhprof.output_dir')."/xhprof_.$id.".out";
    file_put_contents($nameFile, $toWrite, FILE_APPEND | LOCK_EX );
}
```

```

if (isset( $_COOKIE['XHPROF_PROFILE']) ) {
    setcookie("XHPROF_PROFILE", "", time()-3600);
}

```

3.7.5 Configuration uprofiler for swingScaViewer

Uoprofiler is a fork of xhprof, so the configuration of uprofiler are similar to those of xhprof. The files listed below, must be uploaded on the server before the tests. The files ./phptools/*.php must be visible from DocumentRoot of the application with a URI like /phptools/*.php

uprofiler.ini

uprofiler.ini to be included or a paragraph of **php.ini**

Filters run correctly with uprofiler.

But I fail to found a correct dll for Windows and Wamp Server, PHP 5.5.12, it runs correctly with Linux.

```

[xhprof]
extension = "uprofiler.so"
uprofiler.output_dir = "/var/tmp/xhprof"
; For outputs to be parsed by swingScaViewer use these scripts :
auto_prepend_file = "/var/www/html/phptools/uprofiler_header_scripts.php"

; For outputs to be parsed by native tools coming with uprofiler, use these scripts :
;auto_prepend_file = "/var/www/html/phptools/uprofiler_header_natif.php"

;
; And to get the 2 kind of file output
;auto_prepend_file = "/var/www/html/phptools/uprofiler_header_mixed.php"

```

uprofiler_header_scripts.php

```

<?php

if (extension_loaded('uprofiler')) {

    // see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;

    if (mustProfile($_SERVER['PHP_SELF']))
    {
        $needCookie=withCookie($_SERVER['PHP_SELF']) ;
        if ( ! $needCookie || (( $needCookie) && (isset($_COOKIE['UPROFILER_PROFILE'])))){
            // set the RAND variable in file phptools_incl_commun.php to the frequency of
profiling that you want; 10 means : 1 profiled script each 10 runs.

            $tirage = rand(1,RAND);
            if ($tirage === 1 ) {
                if( SUIVIPROF === true )      {
                    $t = microtime(true);
                    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                    $date_format=$date->format ( "Y-m-d:H:i:s.u" );

```

```

$nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler.suivi.out";
$treat=$date_format.";Begin treatment of ".
$_SERVER['PHP_SELF'].";\n";
file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
}
register_shutdown_function('footer_scripts');
uprofiler_enable(UPROFILER_FLAGS_CPU | UPROFILER_FLAGS_MEMORY);

}

}

}

function footer_scripts() {
$uprofiler_data = uprofiler_disable();
$datastr=print_r($uprofiler_data, true);
$id=$id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", " ", " " ,
$_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
$t = microtime(true);
$micro = sprintf("%06d", ($t - floor($t)) * 1000000);
$date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
$date_format=$date->format ( "Y-m-d:H:i:s.u" );
$firstLine = "## NEW PROFILE FILE BEGIN\n";
$lastLine="## NEW PROFILE FILE END=".$date_format."\n";
if( $SUIVIPROF === true ) {
$nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler_suivi.out";
$treat=$date_format.";End treatment of ". $_SERVER['PHP_SELF'].";\n";
file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
}
$towrite=$firstLine."cmd=".$_SERVER['DOCUMENT_ROOT']. $_SERVER['PHP_SELF']."\n".$datastr."\n".
$lastLine."\n";
$nameFile=ini_get('uprofiler.output_dir')."/uprofiler_". $id ."_.out";
file_put_contents($nameFile, $towrite, FILE_APPEND | LOCK_EX );
if (isset( $_COOKIE["UPROFILER_PROFILE"])) {
setcookie("UPROFILER_PROFILE", "", time()-3600);
}
}
}

```

phptools_incl_commun.php

This file is the same used by the three tools : xdebug/xhprof/uprofiler
In red, the configurations to set (inclusive filter, exclusive filter, randomization).
Read the comments at the beginning of the file for the how-to

```

<?php
// set array of pattern of wanted scripts to profile
    // can take 2 forms for value :
        // - withCookie => it is profiled if a cookie XHPROF_PROFILE is present and $tirage === 1
        //      - noCookie => always profiled if $tirage === 1

; an array of including filter to be set
define ('PATTERNS',serialize( array(
    '/phpmyadmin/index.php' => 'noCookie',
    'testjlp' => 'noCookie'
), 'white' => 'noCookie'
//      , 'version_check.php' => 'noCookie'
)));

// Some file containing AJAX or JS scripts may not run correctly with xhprof, you can exclude them:
// key is just a mnemonic, the value is the pattern to exclude here
; an array of excluding filter to be set
define ("EXCLUDED" , serialize (array ('javascript' => 'js'))));

```

```
//define ("EXCLUDED" , serialize (array ()));
// Sampling one script every rand.
// rand = 10 means profile 1 script every 10 runs of the script.
// to always profile a script set RAND to "1"
define ('RAND' , '1');

// to follow the execution of xhprof/uprofiler
// gives a little level of debug
define ('SUIVIPROF' , false);

// truncated : defining functions below.
...
```

uprofiler_header_natif.php

```
<?php

if (extension_loaded('uprofiler')) {
    // see comments in included file phptools_incl_commun.php
    include 'phptools_incl_commun.php';

    if (mustProfile($_SERVER['PHP_SELF'])) {
        $needCookie=withCookie($_SERVER['PHP_SELF']);
        if ( ! $needCookie || (($needCookie) && (isset($_COOKIE['UPROFILER_PROFILE'])))) {

            $uprofiler_data = uprofiler_disable();
            $datas=print_r($uprofiler_data, true);
            if (strlen($datas) > 10) {
                $id=$id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", "_",
, $_SERVER['DOCUMENT_ROOT'] . $_SERVER['PHP_SELF'])));
                $t = microtime(true);
                $micro = sprintf("%06d", ($t - floor($t)) * 1000000);
                $date = new DateTime( date('Y-m-d H:i:s.' . $micro, $t) );
                $date_format=$date->format ("Y-m-d:H:i:s.u");
                $firstLine = "## NEW PROFILE FILE BEGIN\n";
                $lastLine="## NEW PROFILE FILE END=".$date_format."\n";
                if( SUIVIPROF === true )      {

                    $nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler_suivi.out";
                    $treat=$date_format.";End traitement of ".
$_SERVER['PHP_SELF'].";\n";
                    file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                }
                $towrite=$firstLine."cmd=".$_SERVER['DOCUMENT_ROOT'].
$_SERVER['PHP_SELF']."\n".$datas."\n".$lastLine."\n";
                $nameFile=ini_get('uprofiler.output_dir')."/uprofiler_". $id ." .out";
                file_put_contents($nameFile, $towrite, FILE_APPEND | LOCK_EX );
            }
        }
        if (isset( $_COOKIE["UPROFILER_PROFILE"] ) ) {
            setcookie("UPROFILER_PROFILE", "", time()-3600);
        }
    }
}
```

uprofiler_footer.php

```
<?php
```

```

if (extension_loaded('uprofiler')) {

    // see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;

    if (mustProfile($_SERVER['PHP_SELF']))
    {
        $needCookie=withCookie($_SERVER['PHP_SELF']) ;
        if ( ! $needCookie || (( $needCookie) && (isset($_COOKIE['UPROFILER_PROFILE'])))){
            // set the RAND variable in file phptools_incl_commun.php to the frequency of
profiling that you want; 10 means : 1 profiled script each 10 runs.

            $tirage = rand(1,RAND);
            if ($tirage === 1 ) {
                if( SUIVIPROF === true )
                {
                    $t = microtime(true);
                    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                    $date_format=$date->format ( "Y-m-d:H:i:s.u" );

                    $nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler.suivi.out";
                    $treat=$date_format.";Begin treatment of ".
$_SERVER['PHP_SELF'].";\n";
                    file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                }
                register_shutdown_function('footer_natif');
                uprofiler_enable(UPROFILER_FLAGS_CPU | UPROFILER_FLAGS_MEMORY);

            }
        }
    }
}

function footer_natif() {
    $uprofiler_data = uprofiler_disable();
    $datas=print_r($uprofiler_data, true);
    // Generation file for native uprofiler visualisation tools
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/uprofiler_lib/utils/uprofiler_lib.php";
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/uprofiler_lib/utils/uprofiler_runs.php";
    $uprofiler_runs = new uprofilerRuns_Default();
    $id=str_replace(":", "_", str_replace(".", "_", str_replace ("/", "_",
$_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
    $run_id= $uprofiler_runs->save_run($uprofiler_data, $id);
    if( SUIVIPROF === true )
    {
        $t = microtime(true);
        $micro = sprintf("%06d",($t - floor($t)) * 1000000);
        $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
        $date_format=$date->format ( "Y-m-d:H:i:s.u" );
        $nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler_suivi.out";
        $treat=$date_format.";End treatment of ". $_SERVER['PHP_SELF'].";\n";
        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
    }
    if (isset( $_COOKIE["UPROFILER_PROFILE"] ) ) {
        setcookie("UPROFILER_PROFILE", "", time()-3600);
    }
}

```

uprofiler_header_mixed.php

```
<?php

if (extension_loaded('uprofiler')) {

    // see comments in included file phptools_incl_commun.php
    include_once 'phptools_incl_commun.php' ;

    if (mustProfile($_SERVER['PHP_SELF']))
    {
        $needCookie=withCookie($_SERVER['PHP_SELF']) ;
        if ( ! $needCookie || ((($needCookie) && (isset($_COOKIE['UPROFILER_PROFILE'])))) {
            // set the RAND variable in file phptools_incl_commun.php to the frequency of
profiling that you want; 10 means : 1 profiled script each 10 runs.

            $tirage = rand(1,RAND);
            if ($tirage === 1 ) {
                if( SUIVIPROF === true )      {
                    $t = microtime(true);
                    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
                    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
                    $date_format=$date->format ( "Y-m-d:H:i:s.u" );

                    $nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler.suivi.out";
                    $treat=$date_format.";Begin treatment of ".
$_SERVER['PHP_SELF'].";\n";
                    file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX
);
                }
            }
        }
    }

function footer_mixed() {
    $uprofiler_data = uprofiler_disable();
    $datas=print_r($uprofiler_data, true);
    // Generation file for native uprofiler visualisation tools
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/uprofiler_lib/utils/uprofiler_lib.php";
    include_once $_SERVER['DOCUMENT_ROOT'] . "/phptools/uprofiler_lib/utils/uprofiler_runs.php";
    $uprofiler_runs = new uprofilerRuns_Default();
    $id=str_replace(":", "_", str_replace(".", "_", str_replace ("/" , "_" ,
$_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
    $run_id= $uprofiler_runs->save_run($uprofiler_data, $id);
    // Generation file for tool swingScaViewer
    $id=$id=str_replace(":", "_", str_replace(".", "_", str_replace ("/" , "_" ,
$_SERVER['DOCUMENT_ROOT'] .$_SERVER['PHP_SELF'])));
    $t = microtime(true);
    $micro = sprintf("%06d",($t - floor($t)) * 1000000);
    $date = new DateTime( date('Y-m-d H:i:s.'.$micro,$t) );
    $date_format=$date->format ( "Y-m-d:H:i:s.u" );
    $firstLine = "## NEW PROFILE FILE BEGIN\n";
    $lastLine="## NEW PROFILE FILE END=".$date_format."\n";
    if( $suiviprof === true ) {
        $nameFileSuivi=ini_get('uprofiler.output_dir')."/uprofiler_suivi.out";
        $treat=$date_format.";End treatment of". $_SERVER['PHP_SELF'].";\n";
        file_put_contents($nameFileSuivi, $treat, FILE_APPEND | LOCK_EX );
    }
    $toWrite=$firstLine."cmd=".$_SERVER['DOCUMENT_ROOT'].$_SERVER['PHP_SELF']."\n".
$lastLine."\n";
    $nameFile=ini_get('uprofiler.output_dir')."/uprofiler_.$id.".out";
    file_put_contents($nameFile, $toWrite, FILE_APPEND | LOCK_EX );
}

```

```

if (isset( $_COOKIE["UPROFILER_PROFILE"] ) ) {
    setcookie("UPROFILER_PROFILE", "", time()-3600);
}

```

3.7.6 Screens shots



The configuration of other aspects of PHP (Version / Thread Safe, Non Thread Safe, Apache Pre-fork , installing and/or compiling xdebug ...) is out of the scope of this document.

The goal of this Menu is to generate a tree and table views of the results of PHP tools profiling. Three kind of trace are treated.

For the three sub-menus, I tried to have the same kind of visual output (A tree and 2 tables)

The Xdebug Trace with Format 1

The **php.ini** file, or an included file must be configured as is :

```

zend_extension = "c:/opt/wamp/bin/php/php5.5.12/zend_ext/php_xdebug-2.2.5-5.5-vc11-
x86_64.dll"
;Or Equivalent for Linux
; zend_extension = xdebug.so
[xdebug]
xdebug.coverage_enable = off

xdebug.remote_enable = off
xdebug.auto_trace=0
xdebug.trace_enable_trigger=1
xdebug.trace_options = 1

```

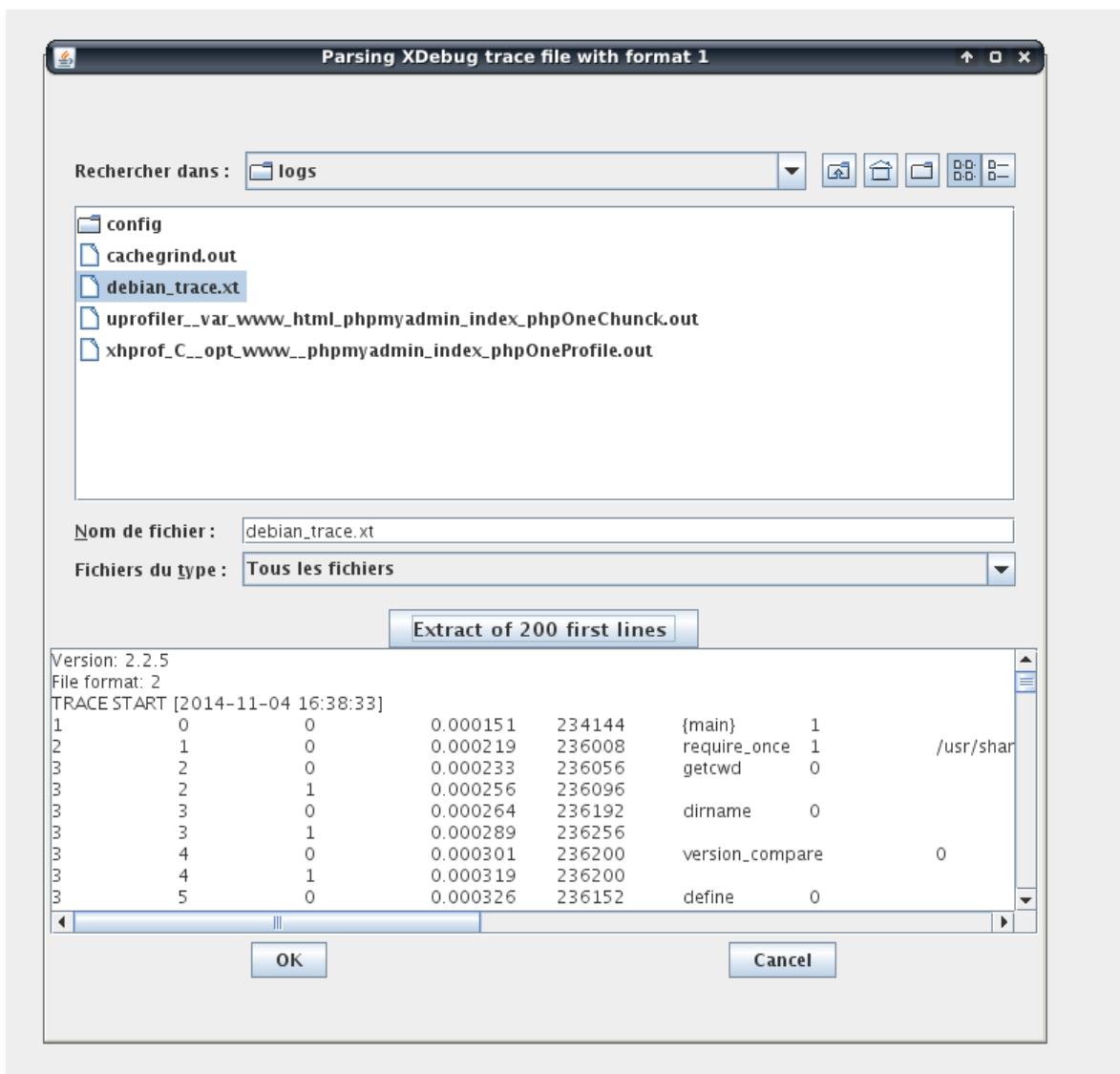
```
xdebug.collect_params=0
xdebug.collect_return=0
xdebug.trace_format=1
xdebug.trace_output_dir="c:/opt/tmp"
xdebug.trace_output_name="trace"
```

The logs to parse [have the format](#) :

```
Version: 2.2.5
File format: 2
TRACE START [2014-11-23 18:09:12]
1      0      0    0.000038    236328 {main}  1
/var/www/html/xhprof_header_scripts.php      0
2      1      0    0.000094    236800 include_once  1
/var/www/html/phptools_incl_commun.php      /var/www/html/xhprof_header_scripts.php      6
3      2      0    0.000110    237248 serialize      0
/var/www/html/phptools_incl_commun.php      13
3      2      1    0.000127    237344
3      3      0    0.000134    237040 define      0
/var/www/html/phptools_incl_commun.php      13
3      3      1    0.000146    237096
3      4      0    0.000152    237304 serialize      0
/var/www/html/phptools_incl_commun.php      17
3      4      1    0.000162    237400
3      5      0    0.000167    237096 define      0
/var/www/html/phptools_incl_commun.php      17
3      5      1    0.000176    237152
3      6      0    0.000181    237056 define      0
/var/www/html/phptools_incl_commun.php      23
3      6      1    0.000190    237056
3      7      0    0.000196    237008 function_exists      0
/var/www/html/phptools_incl_commun.php      26
3      7      1    0.000206    237008
```

In green the index time given in s (precision until microsecond) . In black bold the memory size in bytes

Nota: The Format is really 1 (not 2 as you see in the output)



After parsing the result is like this :

The screenshot shows the SwingScaViewer interface. At the top, there's a menu bar with 'File', 'ScalaLogParser', 'Viewers', 'ScaFileStats', 'MyCommands', 'Tools', 'PHP Tools', and 'ScaViewer Infos'. Below the menu is a toolbar with buttons for 'Current View: Duration', 'Duration View', and 'Memory View'. The main area has three tabs: 'View Tree as Table', 'Concat functions', and 'Drill Down'. The 'View Tree as Table' tab is selected, displaying a hierarchical tree of PHP functions. The 'Concat functions' tab shows a large table of function costs. The 'Drill Down' tab is also visible.

fileNameRow	nbExec	durationInMillis	selfDurationInMillis	memBefore	memAfter	deltaMem
<code>/s/OutputBuffering.class.php_0</code>	1	89.188	0.980	3782312	10520	-3771792
<code>/s/Response.class.php_114</code>	1	89.096	1.073	3780376	10520	-3769856
<code>/s/Response.class.php_364</code>	1	0.020	0.020	3780376	3780376	0
<code>/s/Response.class.php_365</code>	1	0.017	0.017	3780408	3780408	0
<code>/s/Response.class.php_366</code>	1	0.017	0.017	3780408	3780408	0
<code>/s/PMA_Response->response</code>	1	0.016	0.016	3780344	3780344	0
<code>/s/PMA_Response->response</code>	1	86.058	1.208	3780376	3863352	82976
<code>/s/PMA_Header->sendHeaders</code>	1	0.010	0.010	3780472	3871192	90926
<code>/s/PMA_Config->get</code>	1	0.054	0.110	3781088	3782600	55896
<code>/s/PMA_Config->get</code>	1	0.064	0.028	3781088	3781088	0
<code>/s/PMA_Config->get</code>	1	0.019	0.019	3781136	3781136	0
<code>/s/PMA_Config->get</code>	1	0.016	0.016	3781136	3781136	0
<code>/s/PMA_Config->get</code>	1	0.021	0.021	3781232	3781232	96
<code>/s/Header->defined</code>	1	0.017	0.017	3781416	3781416	0
<code>/s/Header->defined</code>	1	0.022	0.022	3781464	3781560	96
<code>/s/Header->defined</code>	1	0.018	0.018	3781564	3782016	280
<code>/s/Header->defined</code>	1	0.058	0.058	3782016	3782248	232
<code>/s/PMA_noCacheHeader->defined</code>	1	0.176	0.055	3782288	3782488	192
<code>/s/Header->defined</code>	1	0.015	0.015	3782392	3782392	0
<code>/s/Header->defined</code>	1	0.015	0.015	3782392	3782440	96
<code>/s/Header->defined</code>	1	0.017	0.017	3782440	3782536	96
<code>/s/Header->defined</code>	1	0.017	0.017	3782536	3782632	96
<code>/s/Header->defined</code>	1	0.016	0.016	3782632	3782600	0
<code>/s/Header->defined</code>	2	0.015	0.015	3782536	3782536	0
<code>/s/Header->defined</code>	2	0.018	0.018	3782456	3782464	8
<code>/s/Header->defined</code>	2	0.016	0.016	3782440	3782440	40
<code>/s/Header->defined</code>	2	0.017	0.017	3782440	3782536	96
<code>/s/Header->defined</code>	2	0.017	0.017	3782536	3782632	96
<code>/s/Header->defined</code>	2	0.016	0.016	3782632	3782600	0
<code>/s/Header->defined</code>	2	0.015	0.015	3782600	3782600	0
<code>/s/Header->defined</code>	2	0.018	0.018	3782600	3782736	136
<code>/s/Header->defined</code>	2	0.014	0.014	3782736	3782736	49

The tree and the first table are synchronized, so you can see both the position of the function in the calls tree, and the corresponding cost values in the first tables.

The second table is the concatenation of the costs of functions of the entire output file.

When right clicking on a function in the first table or in the tree, you open the tab Drill/Down for the selected function :

The screenshot shows the SwingScaViewer interface with the 'Drill Down' tab selected. It displays three tabs: 'Parent Function', 'Current Function', and 'Child Functions'. The 'Parent Function' tab shows the function `/s/OutputBuffering.class.php_0` with a duration of 89.188 ms. The 'Current Function' tab shows the function `/s/Response.class.php_114` with a duration of 89.096 ms. The 'Child Functions' tab shows the child functions of `/s/Response.class.php_114`, including `/s/Response.class.php_364`, `/s/Response.class.php_365`, `/s/Response.class.php_366`, `/s/OutputBuffering->getInstance`, `/s/OutputBuffering->getContents`, `/s/PMA_Response->response`, `/s/Response->_getHttpStart`, and `/s/Header->defined`.

fileNameRow	nbExec	durationInMillis	selfDurationInMillis	memBefore	memAfter	deltaMem
<code>/s/OutputBuffering.class.php_0</code>	1	89.188	0.980	3782312	10520	-3771792
<code>/s/Response.class.php_114</code>	1	89.096	1.073	3780376	10520	-3769856
<code>/s/Response.class.php_364</code>	1	0.020	0.020	3780376	3780408	0
<code>/s/Response.class.php_365</code>	1	0.026	0.026	3780408	3780408	0
<code>/s/Response.class.php_366</code>	1	0.015	0.015	3780408	3780408	0
<code>/s/OutputBuffering->getInstance</code>	1	0.017	0.017	3780392	3780392	0
<code>/s/OutputBuffering->getContents</code>	1	0.016	0.016	3780344	3780344	0
<code>/s/PMA_Response->response</code>	1	86.058	1.208	3780376	3863352	82976
<code>/s/Response->_getHttpStart</code>	1	0.401	0.045	3863360	3863408	48
<code>/s/Header->defined</code>	1	0.035	0.035	3806824	3807096	272

You can also drill/dow by right clicking in the Parent Table or right cliking on a row of the child Table.

Sub-Menu XDebug_profile

The **php.ini** file, or an included file must be configured as is :

```
zend_extension = "c:/opt/wamp/bin/php/php5.5.12/zend_ext/php_xdebug-2.2.5-5.5-vc11-x86_64.dll"
;;Or Equivalent for Linux
; zend_extension = xdebug.so

[xdebug]
xdebug.coverage_enable = Off
xdebug.profiler_enable = off
xdebug.profiler_output_name = "cachegrind.out"
xdebug.profiler_output_dir = "c:/opt/tmp"

xdebug.profiler_enable=0
xdebug.profiler_enable_trigger=1
xdebug.profiler_append = 1
```

The logs to parse have the format cacheGring / valGrind:

```
==== NEW PROFILING FILE =====
version: 1
creator: xdebug 2.2.5
cmd: C:\opt\wamp\apps\phpmyadmin4.1.14\index.php
part: 1
positions: line

events: Time

f1=php:internal
fn=php::getcwd
37 0

f1=php:internal
fn=php::dirname
37 0

f1=php:internal
fn=php::version_compare
45 0

f1=php:internal
fn=php::define
52 0
```

```

fl=php:internal
fn=php::defined
9 0

fl=php:internal
fn=php::defined
9 0

fl=C:\opt\wamp\apps\phpmyadmin4.1.14\libraries\Message.class.php
fn=require_once::C:\opt\wamp\apps\phpmyadmin4.1.14\libraries\Message.class.php
1 0

fl=C:\opt\wamp\apps\phpmyadmin4.1.14\libraries>Error.class.php
fn=require_once::C:\opt\wamp\apps\phpmyadmin4.1.14\libraries>Error.class.php
1 47999
cfl=php:internal
cfn=php::defined
calls=1 0 0
9 0
cfl=C:\opt\wamp\apps\phpmyadmin4.1.14\libraries\Message.class.php
cfn=require

```

The duration are in microseconds.

The operating mode is the same like xdebug trace

After parsing the result is like this :

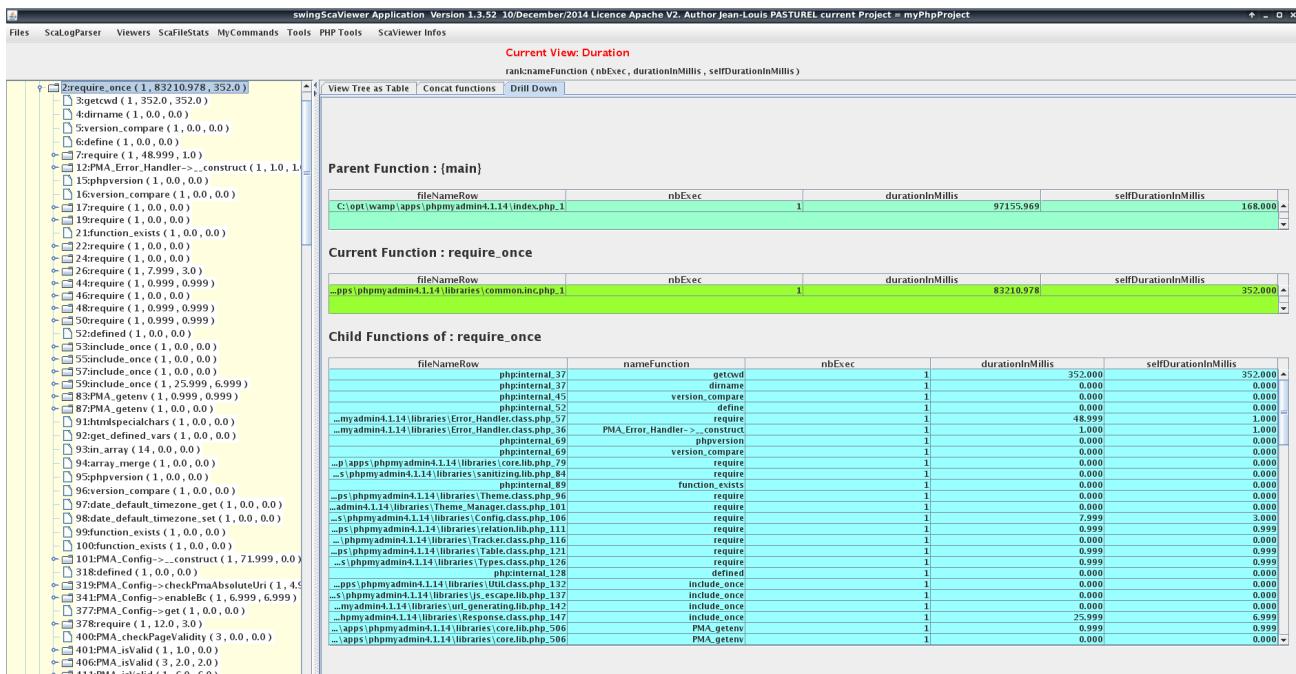
The screenshot shows the SwingScaViewer Application interface. The top menu bar includes items like 'Menu des applications', 'swing...', 'Google...', 'Scala ...', 'etude...', 'manS...', 'swing...', 'Termi...', 'Termi...', 'Termi...', 'docs...', 'apac...', 'logs - ...', 'logs - ...', 'Script...', 'config...', 'Captur...', and '19:18'. Below the menu is a toolbar with icons for 'Files', 'ScalParser', 'Viewers', 'ScaFileStats', 'MyCommands', 'Tools', 'PHP Tools', and 'ScaViewer Infos'. The main window has a title bar 'swingScaViewer Application Version 1.3.52 10/December/2014 Licence Apache V2. Author Jean-Louis PASTUREL current Project = myPhpProject'. The interface is divided into two main sections: a tree view on the left and a table view on the right.

Current View: Duration

rankNameFunction (nbExec, durationInMillis, selfDurationInMillis)					
nameFunction	fileNameRow	nbExec	durationInMillis	selfDurationInMillis	
require_once	_min4.1.14/libraries/common.inc.php	1	83210.978	352.000	352.000
getcwd	phpinternal_37	1	0.000	0.000	0.000
dirname	phpinternal_37	1	0.000	0.000	0.000
version_compare	phpinternal_45	1	0.000	0.000	0.000
define	phpinternal_52	1	0.000	0.000	0.000
require	_4/libraries/Error_Handler/class.php	1	48.999	1.000	1.000
defined	phpinternal_9	1	1.000	1.000	1.000
require_once	_min4.1.14/libraries/Error.class.php	1	47.999	47.999	47.999
defined	phpinternal_9	1	47.999	47.999	47.999
require	_4.1.14/libraries/Messages.class.php	1	0.000	0.000	0.000
require	_4/libraries/Error_Handler/class.php	1	1.000	1.000	1.000
defined	phpinternal_44	1	1.000	1.000	1.000
set_error_handler	phpinternal_45	1	0.000	0.000	0.000
phpversion	phpinternal_69	1	0.000	0.000	0.000
version_compare	phpinternal_69	1	0.000	0.000	0.000
require	_admin4.1.14/libraries/or.lib.php	1	0.000	0.000	0.000
defined	phpinternal_10	1	0.000	0.000	0.000
require	_in4.1.14/libraries/sanitizing.lib.php	1	0.000	0.000	0.000
defined	phpinternal_8	1	0.000	0.000	0.000
function_exists	phpinternal_89	1	0.000	0.000	0.000
require	_4.1.14/libraries/TheManager.class.php	1	0.000	0.000	0.000
defined	phpinternal_8	1	0.000	0.000	0.000
require	_aries/TheManager.class.php	1	0.000	0.000	0.000
defined	phpinternal_8	1	0.000	0.000	0.000

The tree and the first table are synchronized, so you can see both the position of the function in the calls tree, and the corresponding cost values in the first tables.

The second table is the concatenation of the costs of functions of the entire output file.



Sub-Menu Xhprof

To generate the traces, the configuration needs to put and uncompress the file **phptools.zip** (or **phptools.tar.gz** they contain the same files) under the “**DocumentRoot**” of the Web Application (Be care to use the correct DocumentRoot when there are virtual hosts, and also if the rewrite Engine is set in Apache)

The archives **phptools.zip** and **phptools.ta.gz** are in the <swingScaViewer>/uploads directory. The **php.ini** file, or an included file must be configured as is :

```
[xhprof]
extension= "C:/opt/wamp/bin/php/php5.5.12/ext/php_xhprof_ts.dll"
xhprof.output_dir = "C:/opt/tmp"

auto-prepend_file = "C:/opt/www/phptools/xhprof_header_scripts.php"
```

The archive **phptools.zip**, contains the **auto_prepend_file** and **auto_append_file**, for different uses of Xhprof and its fork named uprofiler.

The **prepend_file** allows to filter the scripts php to profile, and also can configure a sampling with a random function.

The **auto_append_file** writes the logs if any.

SwingScaViewer parses the two type of logs (**xhprof** and **uprofiler**) with this sub-menu.

The logs to parse have the format below :

```
## NEW PROFILE FILE BEGIN
cmd=C:/opt/www//phpmyadmin/index.php
Array
(
    [main()=>load:::phpmyadmin4.1.14/index.php] => Array
        (
            [ct] => 1
            [wt] => 2444
            [cpu] => 0
            [mu] => 113880
            [pmu] => 284616
        )
)

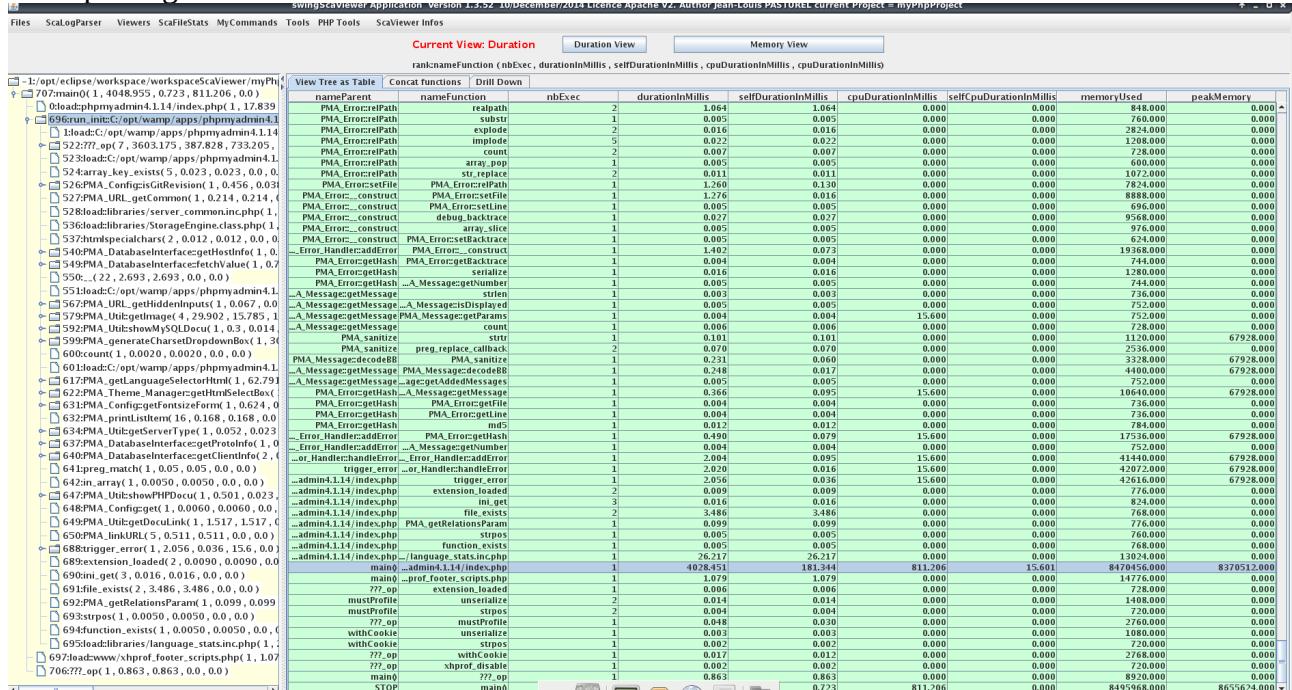
[run_init:::C:\opt\wamp\apps\phpmyadmin4.1.14\index.php=>load:::C:\opt\wamp\apps\phpmyadmin4.1.14\lib
raries\common.inc.php] => Array
(
    [ct] => 1
    [wt] => 2429
    [cpu] => 0
    [mu] => 149360
    [pmu] => 122216
)

[???_op=>getcwd] => Array
(
    [ct] => 1
    [wt] => 9
    [cpu] => 0
    [mu] => 768
    [pmu] => 0
)

)
```

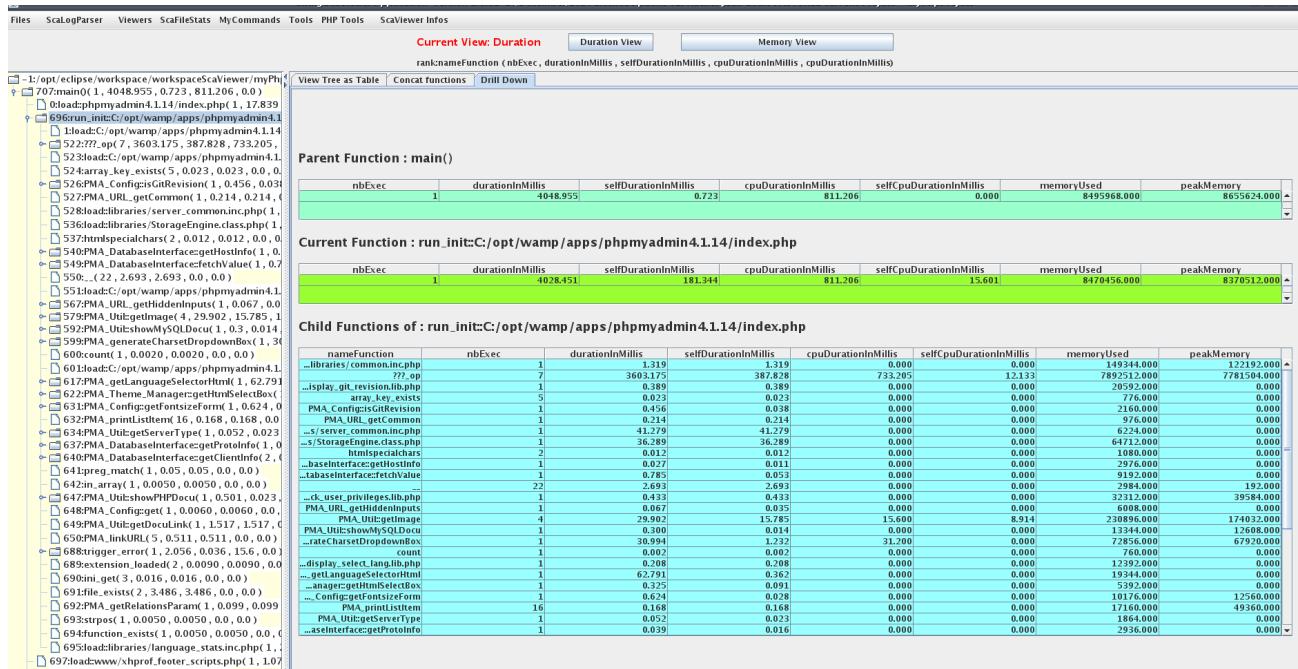
The durations are in microseconds, the memory size in bytes.

After parsing the result is like this:



The tree and the first table are synchronized, so you can see both the position of the function in the calls tree, and the corresponding cost values in the first tables.

The second table is the concatenation of the costs of functions of the entire output file.



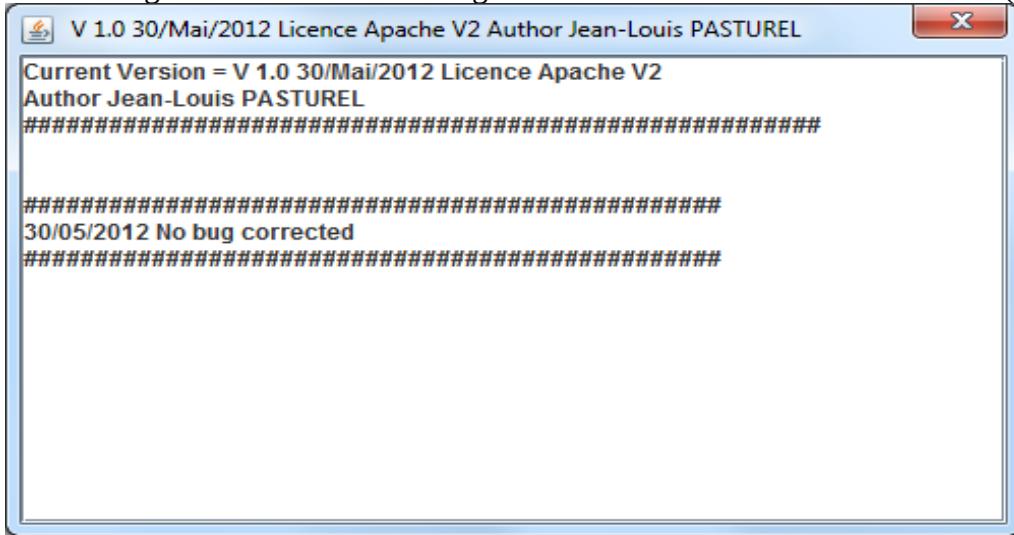
With the same sub-menu, we can parse also files generated by uprofiler:

The drill/down :

swingScalViewer Application Version 1.3.52 10/December/2014 Licence Apache V2. Author Jean-Louis PASTUREL current Project = myPhProject								
Files	ScalParser	Viewers	ScalFileStats	MyCommands	Tools	PHP Tools	ScalViewer Infos	
Current View: Duration Duration View Memory View								
rankNameFunction(nbExec, durationInMillis , selfDurationInMillis , cpuDurationInMillis , cpuDurationInMillis)								
View Tree as Table Concat functions Drill Down								
Parent Function : main()								
nbExec	durationInMillis	selfDurationInMillis	cpuDurationInMillis	selfCpuDurationInMillis	memoryUsed	peakMemory		
1	115.579	0.026	112.000	0.000	4391800.000	4411304.000		
Current Function : run_init:phpmyadmin/index.php								
nbExec	durationInMillis	selfDurationInMillis	cpuDurationInMillis	selfCpuDurationInMillis	memoryUsed	peakMemory		
1	115.475	0.432	112.000	0.000	4378448.000	4399152.000		
Child Functions of : run_init:phpmyadmin/index.php								
nameFunction	nbExec	durationInMillis	selfDurationInMillis	cpuDurationInMillis	selfCpuDurationInMillis	memoryUsed	peakMemory	
loadLibraries/commonInCgi.php	1	0.003	0.003	0.000	0.000	1000.000	4048.000	
run_initLibraries/commonInCgi.php	1	0.004	0.003	0.000	0.000	3996.000	0.000	
loadLibraries/display_git_revision.lib.php	1	0.017	0.017	0.000	0.000	1912.000	0.000	
run_initLibraries/display_git_revision.lib.php	1	0.004	0.003	0.000	0.000	1912.000	0.000	
PMA_ConfigureFontSizeForm	5	0.001	0.001	0.000	0.000	792.000	0.000	
PMA_ConnectToServer	1	0.011	0.011	0.000	0.000	8000.000	0.000	
PMA_getConnect	1	0.049	0.049	0.000	0.000	8000.000	0.000	
loadLibraries/server_common.inc.php	1	0.016	0.016	0.000	0.000	1216.000	0.000	
run_initLibraries/server_common.inc.php	1	0.214	0.036	0.000	0.000	13624.000	0.000	
loadLibraries/StorageEngine.class.php	1	0.023	0.023	0.000	0.000	9280.000	0.000	
run_initLibraries/storageEngine.class.php	1	0.006	0.008	0.000	0.000	9792.000	0.000	
PMA_DatabaseInterfaceGetInfo	1	0.009	0.004	0.000	0.000	3112.000	2816.000	
PMA_DatabaseInterfaceGetKeyValue	1	0.302	0.026	0.000	0.000	3112.000	32200.000	
PMA_Reconnect	1	0.041	0.011	0.000	0.000	0.000	5656.000	
PMA_ReconfigurableTable	1	0.002	0.002	0.000	0.000	872.000	784.000	
PMA_ReconfigurableTable_getHTMLifyNameTables	1	0.008	0.578	0.000	0.000	1000.000	3000.000	
PMA_ReconfigurableTable_getMatch	22	0.008	0.578	0.000	0.000	1000.000	552.000	
loadLibraries/check_user_privileges.lib.php	1	0.016	0.016	0.000	0.000	1000.000	552.000	
run_initLibraries/check_user_privileges.lib.php	1	0.043	0.012	0.000	0.000	6888.000	5840.000	
PMA_URLGetImage	5	0.479	0.213	0.000	0.000	12168.000	134160.000	
loadLibraries/StorageEngine.class.php	17	0.031	0.011	0.000	0.000	7920.000	11344.000	
PMA_URL_getDatabaseInputs	1	0.014	0.014	0.000	0.000	1000.000	5140.000	
PMA_UtchowshMySQLDocu	1	0.190	0.009	4.000	0.000	17160.000	15656.000	
PMA_generateCharDropdownBox	1	12.199	0.781	12.000	0.000	70576.000	77752.000	
PMA_getRelationsParam1	1	0.000	0.000	0.000	0.000	776.000	0.000	
loadLibraries/display_select_lang.lib.php	1	0.029	0.029	0.000	0.000	1008.000	0.000	

3.8 Menu "ScaViewer Infos"

Menu that gives the Version of SwingScaViewer and the historic of evolution (bugs and features)



4 General Procedure

The procedure describes how to treat a dated log (Apache access logs, Was logs, Application logs , log4J logs, JMeter logs and traces, ...).

The method is :

1. statically parse with the tool ScaFileStats
2. export the result in csv format under the logs folder of the scenario
3. configure ScaLogParser (from scratch or from an existing template)
4. configure the Pivot Panel of ScaLogParser importing the csv file saved in point 2

Below the screen-shots of the method for the file **concat_WAS1.log.gz** (23 Mo zipped, 440 Mo unzipped, 4 500 000 rows)

1. statically parse with the tool ScaFileStats

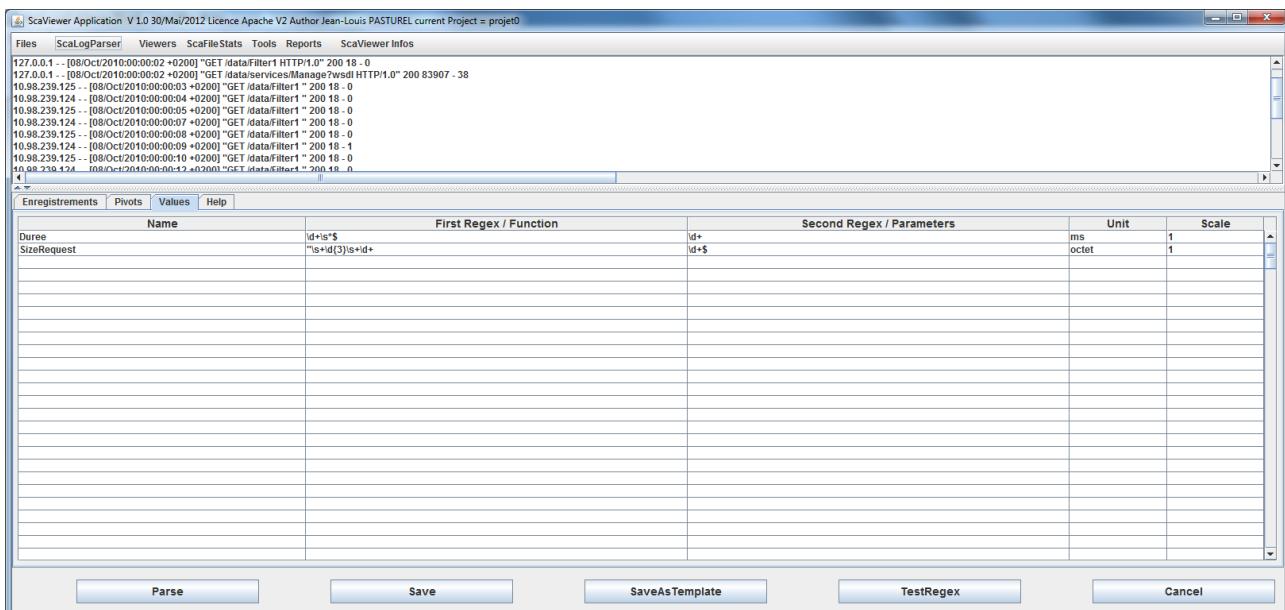
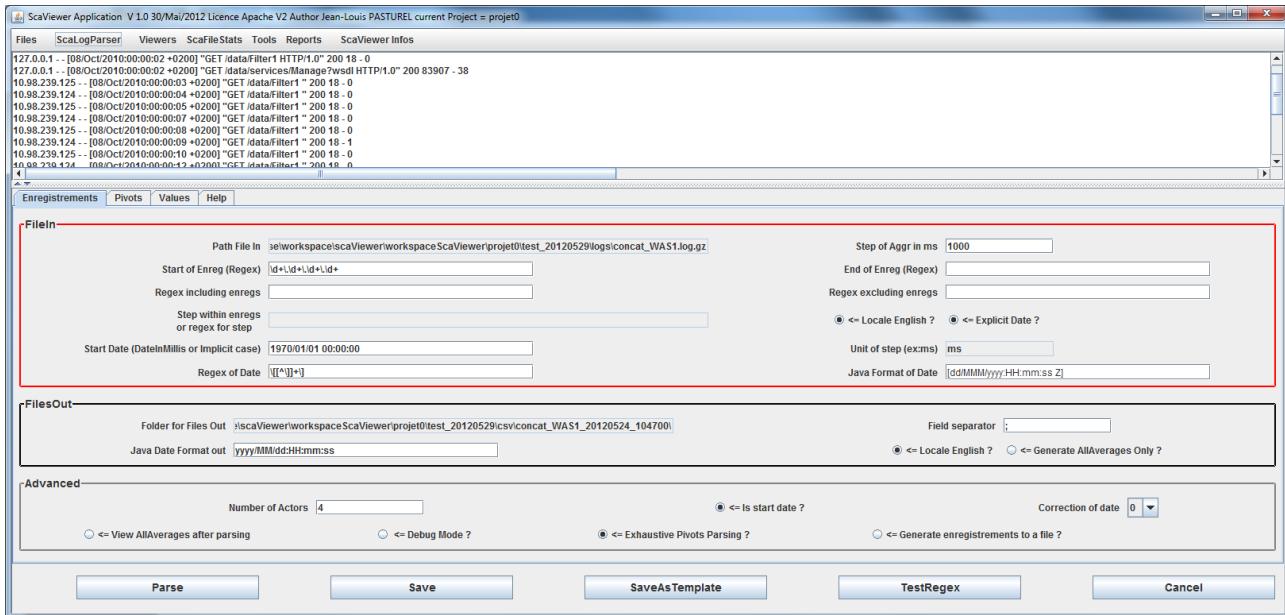
The screenshot shows the 'Advanced analyze of data in a text File' dialog. It includes fields for 'File chosen' (D:\eclipse\workspace\scaViewer\workspaceScaViewer\projet0\test_20120529\logs\concat_WAS1.log.gz), 'Beginning Of Analysis' (08/Oct/2010:00:00:02 +0200), 'End Of Analysis' (26/Jan/2011:04:54:37 +0100), 'Percentile (0<per<100)' (90), 'CSV Separator' (,), 'Number of Actors' (4), and various regex patterns for filtering. Below the dialog is a log output window showing parsing statistics and a pivot table grid.

Num Row	Criteria	Count	Percent	Sum	Average	Minimum	Maximum	Median	Percentile ...	StdDev
0	POST /data/services/Manage	2516352	55.557	369751344	146.939	6	25723	22	181	639.56
1	GET /data/filter1	1665240	36.766	248832	0.149	0	213	0	1	0.901
2	POST /data/services/manageCommercialCustomers	138384	3.055	4284192	30.959	17	6511	27	40	86.567
3	POST /data/services/ManageRole	99552	2.198	1675560	16.831	12	124	14	24	6.484
4	POST /data/services/manageIdentAppli	99504	2.197	801024	8.05	5	64	7	10	3.396
5	GET /data/services/Manage?wsdl	10296	0.227	388704	37.753	29	776	35	38	36.592
6	Total	4529328	100	377149656	83.268	0	25723	18	94	482.269

2. export the result in csv format under the logs folder of the scenario

The screenshot shows a 'Save as template' dialog with a preview of the pivot table and an 'Enregistrer' (Save) dialog. The 'Enregistrer' dialog shows the save location as 'logs' and the file name as 'jonas2.csv'. A preview pane on the left shows the pivot table data.

3. configure ScaLogParser (from scratch or from an existing template)



4. configure the Pivot Panel of ScaLogParser importing the csv file saved in point 2

The screenshot shows the SwingScaViewer application interface. It consists of two main windows.

Top Window:

- Menu bar: Files, ScaLogParser, Viewers, ScafileStats, Tools, Reports, ScaViewer Infos.
- Log pane: Shows log entries from 127.0.0.1 and 10.98.239.125.
- Toolbar: Enregistrements, Pivots, Values, Help.
- Table: Three columns: Name, First Regex, Second Regex.
- File Selection Dialog (Ouvrir): Shows a file browser with 'logs' selected. Files listed: config, jonas.csv, jonas2.csv. Selected file: jonas2.csv. Type: Config Pivots from Files.
- Buttons: Parse, Save, SaveAsTemplate, TestRegex, Cancel.

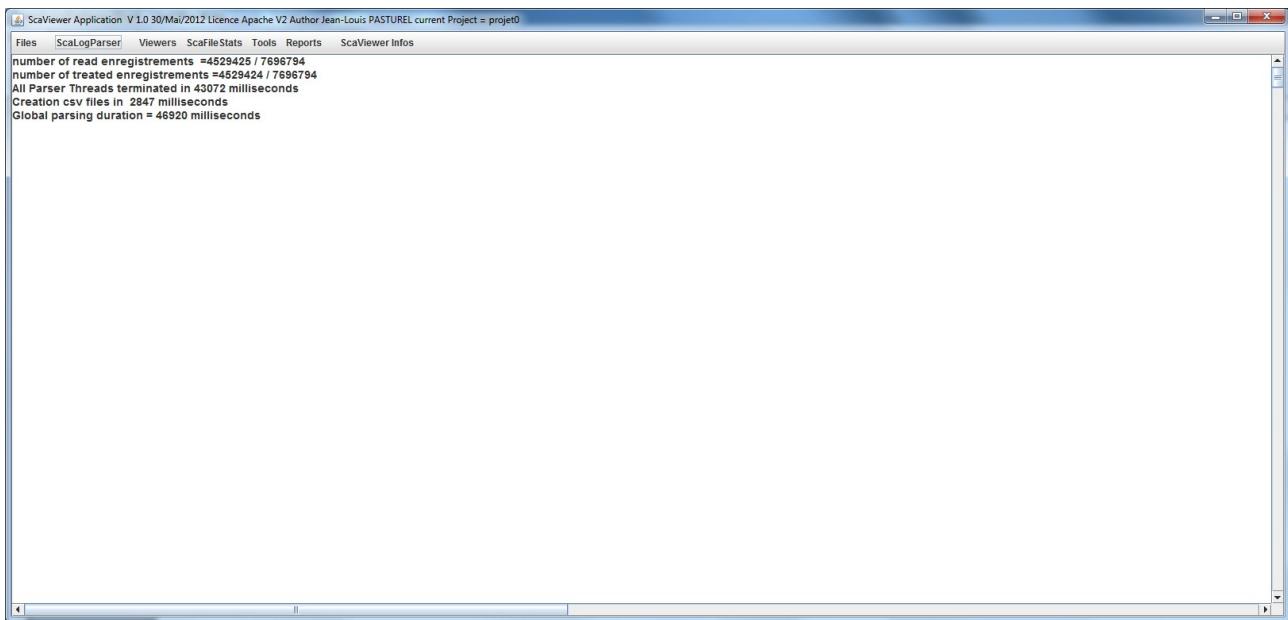
Bottom Window:

- Menu bar: Files, ScaLogParser, Viewers, ScafileStats, Tools, Reports, ScaViewer Infos.
- Log pane: Shows log entries from 127.0.0.1 and 10.98.239.125.
- Toolbar: Enregistrements, Pivots, Values, Help.
- Table: Three columns: Name, First Regex, Second Regex.
- Data in the table:

Name	First Regex	Second Regex
POST_data_services_Manage	POST /data/services/Manage	
GET_data_filter1	GET /data/filter1	
POST_data_services_manageCommercialCustomers_	POST /data/services/manageCommercialCustomers	
POST_data_services_ManageRole	POST /data/services/ManageRole	
POST_data_services_manageIdentAppli_	POST /data/services/manageIdentAppli	
GET_data_services_Manage_wsdl	GET /data/services/Manage?wsdl	

- Buttons: Parse, Save, SaveAsTemplate, TestRegex, Cancel.

and parse :



and view :

5 Annexe

5.1 Examples of Java/Perl regex

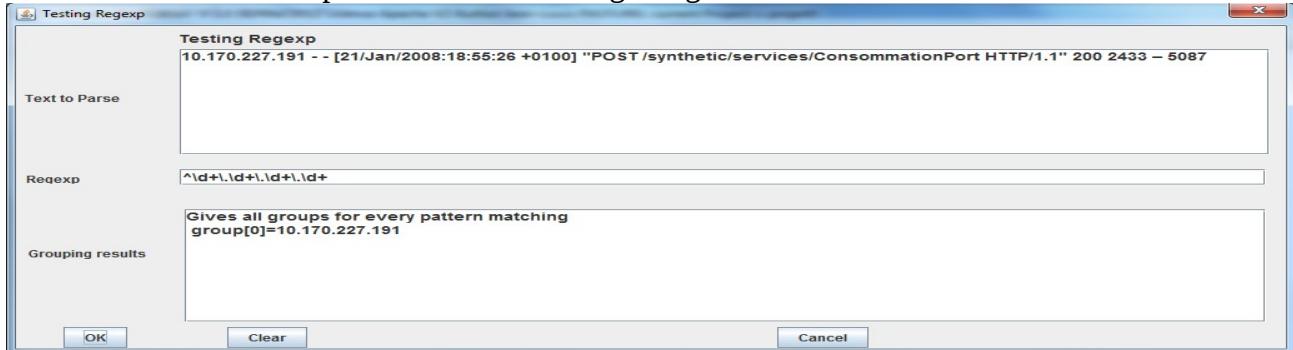
Below some examples of regex to extract datas from a line:

```
10.170.227.191 - - [21/Jan/2008:18:55:26 +0100] "POST /synthetic/services/ConsommationPort
HTTP/1.1" 200 2433 - 5087
```

5.1.1 Regexp : `^\d+\.\d+\.\d+\.\d+`

This regexp reads the line from the beginning of the line (character `^`) , expects one or several numbers until a dot (`\d+\.`) , this repeated three times, and finally expects one or several more numbers (`\d+`).

The dot character is escaped because it has a regex significance

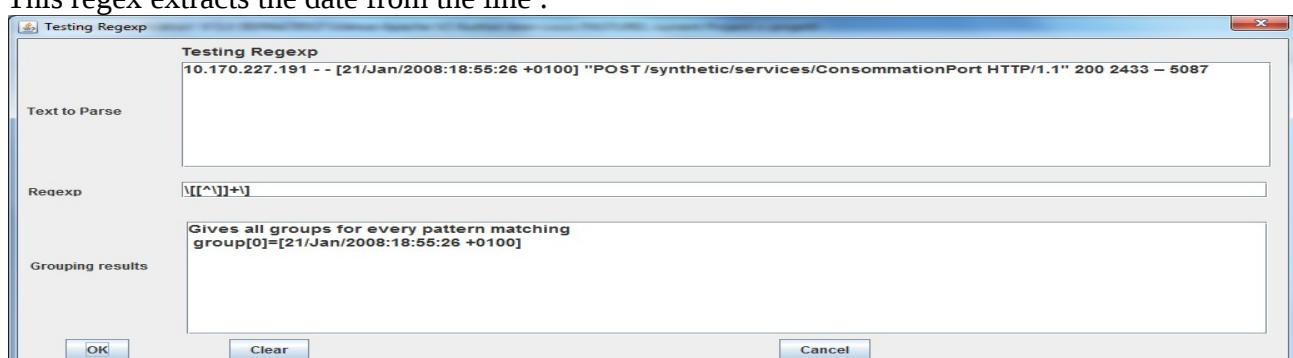


so you extract the IP address at the beginning of the line.

5.1.2 Regexp : `\\[[^\\]]+\\]`

This regexp, reads from the first opening square bracket (`\[`) that must be escaped because it has a regex significance , and after it takes all characters that **are not** closing square bracket, note that in this context, the `^` character is the negation operator when it is not the first character of the regex=> `([^\\]]+)` , and the matching is terminated by the closing square bracket (`\]`)

This regex extracts the date from the line :

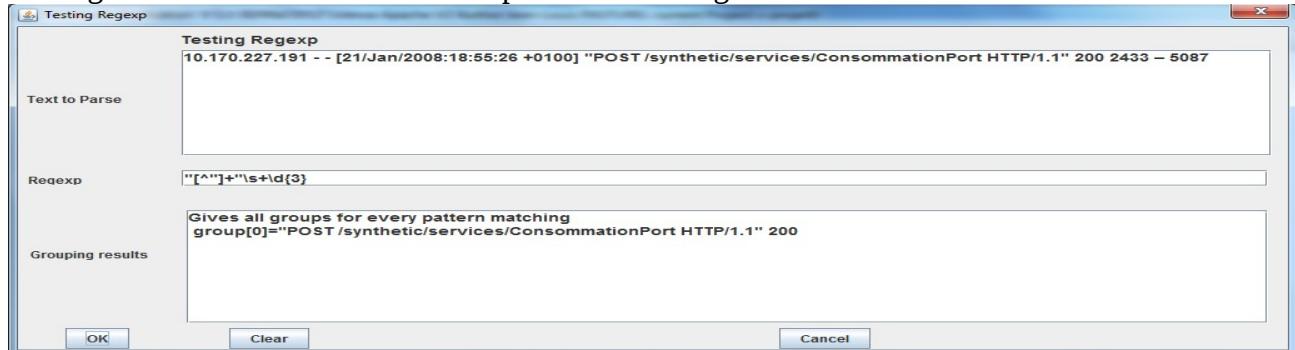


Note : Be care of the difference of signification of the ^ character when it is the first character of the regex, and when it is somewhere else.

5.1.3 Regexp : "[^"]+"\\s+\\d{3}

That must be read from the first double quotation marks ("") and after all characters that are not double quotation marks ([^"]+), after a double quotation mark ("") , after one or more space or tab characters (\s+), and terminated by 3 digits (\d{3}) .

this regex extracts the URL and the http status of the regex from the line :



5.1.4 Regexp : \\s+\\d+\\$

This regexp reads beginning from the end of the line (last character is designated by \$), before the end of the line one or more digits (\d+) and before one or more space or tab characters (\s+).

this regex extracts the last number in the line, for this Apache logs, it corresponds to the duration of the request in micros-seconds.



5.2 Use of key word “function” in swingScaViewer/scaLogParser

5.2.1 File to treat

Extract of the file to treat

```
2008/01/27 17:49:40 val1= 10.3 val2=15
2008/01/27 17:49:40 val1= 10.3 val2=15
```

```
2008/01/27 17:49:40 val1= 10.3 val2=15
2008/01/27 17:49:40 val1= 10.3 val2=15
2008/01/27 17:49:41 val1= 10.3 val2=16
2008/01/27 17:49:41 val1= 10.3 val2=16
2008/01/27 17:49:41 val1= 10.3 val2=16
2008/01/27 17:49:42 val1= 10.3 val2=17
2008/01/27 17:49:42 val1= 10.3 val2=17
2008/01/27 17:49:43 val1= 10.3 val2=18
2008/01/27 17:49:46 val1= 10.3 val2=19
2008/01/27 17:49:47 val1= 10.3 val2=19
2008/01/27 17:49:47 val1= 10.3 val2=19
2008/01/27 17:49:48 val1= 10.3 val2=20
2008/01/27 17:49:48 val1= 10.3 val2=16
2008/01/27 17:49:48 val1= 10.3 val2=16
2008/01/27 17:49:48 val1= 10.3 val2=16
```

Note the space after val1= .

The goal of the function is to get the result of the 4 operations (+,-,*,/) between val1 and val2

5.2.2 Classes Scala "myPlugins"

Below the source of **ConcCompute2Values** class :

```
- ConcCompute2Values.scala
class ConcAdd2Values {
def metInit(tab:Array[String]=null) {
    // To reinitialise static variable if necessary
    // Nothing to do here
}

    /**
     *tabStr(0) is the record to be treated. Afterwards the regex by tuple of 2
items for an extraction in two phases
     * regex tabStr(1) and tabStr(2) to extract the first value
     * regex tabStr(3) and tabStr(4) to extract the second value
     * tabStr(5) is the operand ("+","-","*","/")
     * @param tabStr
     * @return
    */
def retour(tabStr:Array[String]):Double=
{
    var retour[Double.NaN]

    // extract first value
    var regex1=tabStr(1).r
    var ext1=regex1.findFirstIn(tabStr(0))
    var val1=0D
    var val2=0D
    if (None != ext1) {
        val regex2=tabStr(2).r
        val ext2=regex2.findFirstIn(ext1.get)
        if (None != ext2) {
            val1 = ext2.get.toDouble
```

```

        } else {
            return Double.NaN
        }
    } else {
        return Double.NaN
    }

// extract second value
regex1=tabStr(3).r
ext1=regex1.findFirstIn(tabStr(0))

if (None != ext1) {
    val regex2=tabStr(4).r
    val ext2=regex2.findFirstIn(ext1.get)
    if (None != ext2) {
        val2 = ext2.get.toDouble
    } else {
        return Double.NaN
    }
} else {
    return Double.NaN
}

// return the result of the operation
tabStr(5) match {
    case "+" => val1+val2
    case "-" => val1-val2
    case "/" => if (val2 !=0) val1/val2 else Double.NaN
    case "*" => val1 * val2
    case _ => Double.NaN
}
}

}
}

```

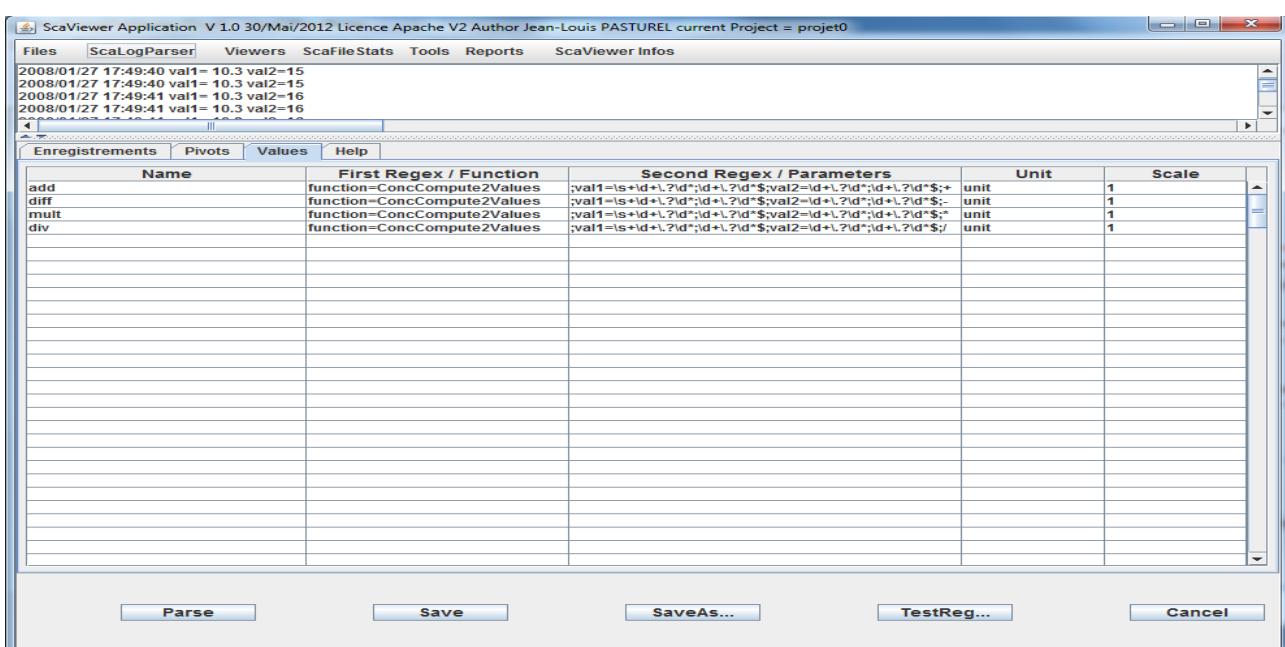
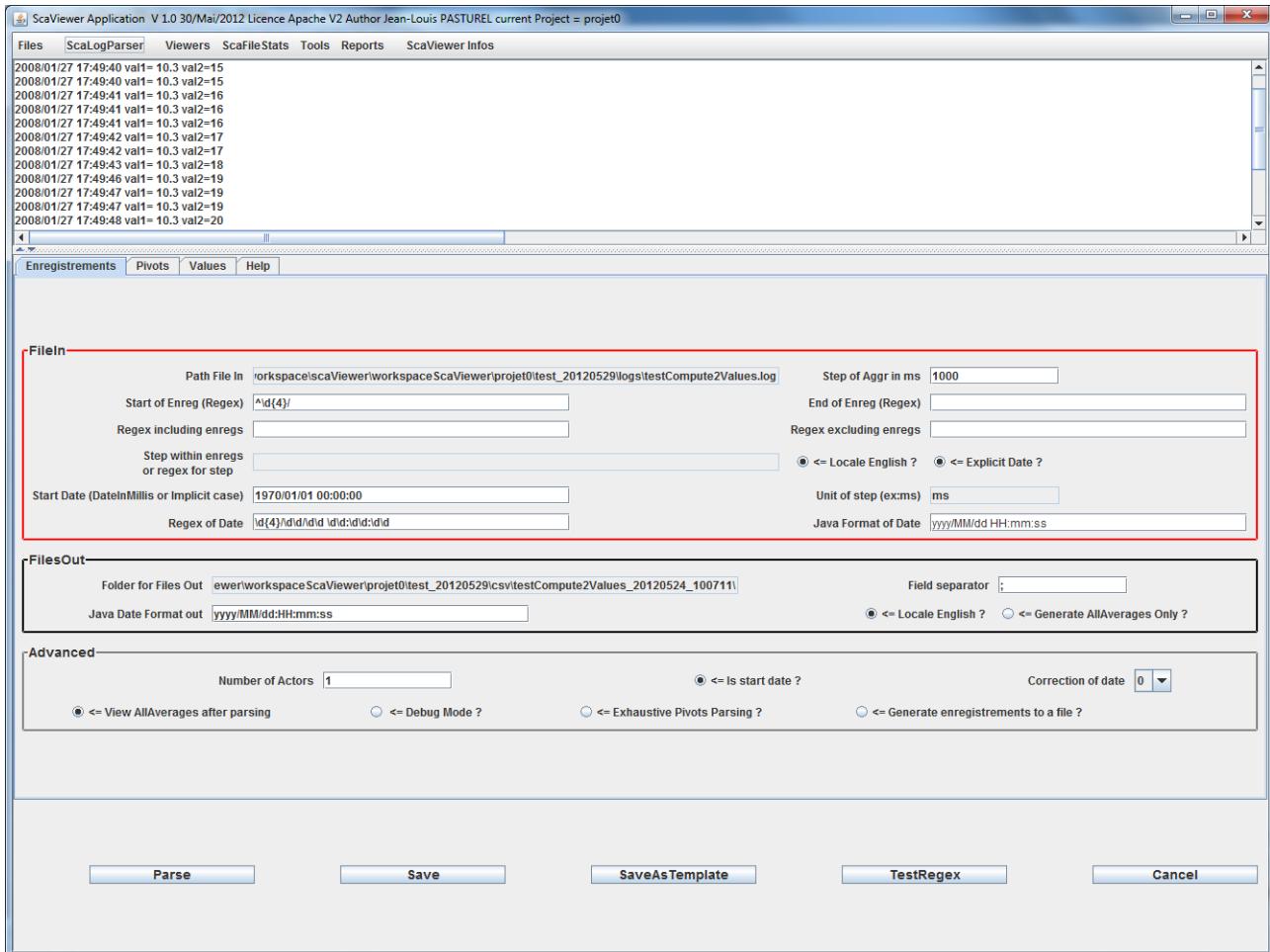
Note : the Conc prefix of the class means that this class can be used in multi threaded parsing (nb Actors >1)

The Scala code of this class is in the directory <**swingScaViewer_Home**>/myPlugins .

The binary of this class is in the jar archive <**swingScaViewer_Home**>/myPlugins/myPlugins.jar

5.2.3 Configuration of swingScaViewer/ScaLogParser

The screens below show the configuration to parse this file :



5.2.4 Visualization

After clicking on Parse, the allAverages are graphed as shown below

