

JASMINe Monitoring

Users guide

v1.3

JASMiNe Monitoring: Users guide

by

Publication date \$Id\$

Copyright © 2008-2011 Bull SAS



This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way Stanford, California 94305 USA

Table of Contents

1. Introduction	1
1.1. JASMINe Monitoring functions	1
1.2. Technologies managed by JASMINe Monitoring	1
2. Getting started	3
2.1. Quick start	3
2.2. Installing JASMINe Monitoring	4
2.2.1. Prerequisites	4
2.2.2. Install JASMINe Monitoring	4
2.2.3. Start/Stop JASMINe Monitoring	6
2.2.4. Uninstall JASMINe Monitoring	7
2.2.5. Manual installation	7
2.3. EoS (Eye of SOA) - The web console	7
2.3.1. Console features	7
3. Working with JASMINe Monitoring	9
3.1. Collect monitoring data	9
3.1.1. Collecting data with Probe Manager	9
3.1.2. Managing probes	11
3.1.3. Managing targets	13
3.1.4. Managing outputs	13
3.1.5. Start collects at JASMINe Monitoring startup	14
3.2. Monitoring	15
3.2.1. Simple monitoring	15
3.2.2. Advanced monitoring	16
3.2.3. Chart navigation	19
3.2.4. Post-mortem analysis	19
3.3. Management of Monitoring Data	19
3.3.1. Using the Web Console	19
3.3.2. Purge Command	20
3.4. Notification	20
3.4.1. Autonomous & Notification editor	20
3.4.2. Notification board	23
4. Advanced features	25
4.1. Manual installation	25
4.1.1. Requisites	25
4.1.2. Configuration	25
4.1.3. Installing JASMINe Monitoring	26
4.1.4. Starting JASMINe Monitoring	26
4.2. Administrate JASMINe Monitoring	26
4.2.1. Managing users and role	26

List of Figures

2.1. JASMINe EoS home page	8
3.1. JASMINe EoS Console with probe manager icon	9
3.2. Probe manager overview	10
3.3. Probe creation popup	11
3.4. Probe detail in edition	12
3.5. Probe detail	12
3.6. Target Management	13
3.7. Output Management	14

Chapter 1. Introduction

1.1. JASMINe Monitoring functions

JASMINe Monitoring offers functionalities to supervise distributed middleware infrastructure.

With JASMINe Monitoring, you will be able to

- Collect monitoring data.
- Store them in databases or in log files.
- Visualize graphs in monitoring dashboards.
- Create notification rules.
- Receive notifications by mail.
- Visualize notifications in a Notification Board.
- Create responses to notifications throwable manually or automatically.

1.2. Technologies managed by JASMINe Monitoring

JASMINe Monitoring targets a lot of servers and platforms :

- Java EE servers and clusters :
 - JOnAS 5.x
 - JOnAS 4.x
 - Apache Tomcat
 - Glassfish
 - JBoss
 - WebLogic
 - ...
- Lightweight ESB :
 - Camel
 - Mule
- Virtualized platforms
- Java middlewares :
 - Orchestra (orchestration)
 - Joram (messaging)

- ...
- Java applications (through JMX)

Chapter 2. Getting started

2.1. Quick start

This section will show you how to quickly install and monitor a server with JASMINe Monitoring.

1. Download the JASMINe Monitoring installer [<http://download.forge.objectweb.org/jasmine/jasmine-monitoring-installer-izpack-full-1.3.1-M2-standard.jar>].

2. Start the installer

```
java -jar jasmine-monitoring-installer-full.jar
```




Note

This requires Java (JDK or JRE) ≥ 1.5 to be installed on your computer (<http://www.oracle.com/technetwork/java/index.html> Software Downloads - JavaSE)

3. In the installer, only set the installation directory. Let other options by default.



4. An entry named OW2 JASMINe has been added to the list of applications installed on your system. To start JASMINe Monitoring, choose  *Start JASMINe monitoring*

5. Open the JASMINe Monitoring console with



JASMINe EoS (web console)

Use *jonas* for both user and password.

6. Open the *Monitoring* module.
7. Load a graph configuration example : select the *jasmine-monitoring-status.xml* file and click the *OK* button.

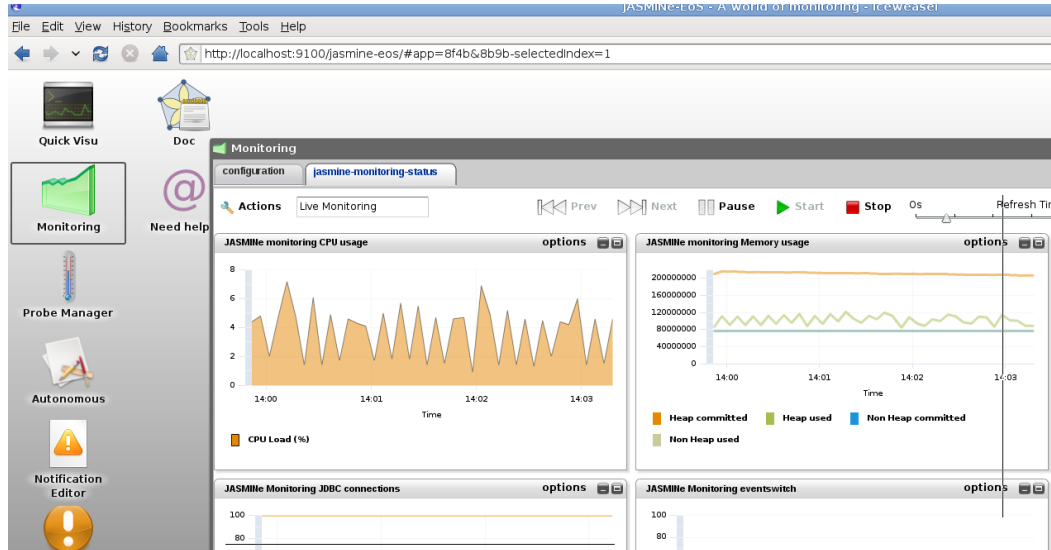
This leads to a new dashboard creation named *jasmine-monitoring-status*.

The new dashboard contains 5 charts:

- JASMINe monitoring CPU usage
- JASMINe monitoring Memory usage
- JASMINe monitoring threads counter

- JASMINe monitoring JDBC connections
- JASMINe monitoring eventswitch

8. Start the monitoring with the button *Start* .



9. What are we monitoring with these charts ?

JASMINe Monitoring is a Java enterprise applications running on JOnAS [<http://wiki.jonas.ow2.org/xwiki/bin/view/Main/WebHome>] Java EE Server.

When JASMINe Monitoring is started, a JOnAS server is started and the JASMINe Monitoring application is deployed on that server.

Moreover, some monitoring probes are started. These probes periodically collect data about the server's resources, such as CPU usage and memory usage.

The charts allow to visualize the monitoring data collected from the server hosting our application.

2.2. Installing JASMINe Monitoring

2.2.1. Prerequisites

- Download the JASMINe Monitoring installer at <http://wiki.jasmine.ow2.org/xwiki/bin/view/Main/Downloads>. The installer is available as an executable jar file or as a Java Web Start application.
- Ensure that you have JDK/JRE 1.5 installed and that your JAVA_HOME environment variable is set.

2.2.2. Install JASMINe Monitoring

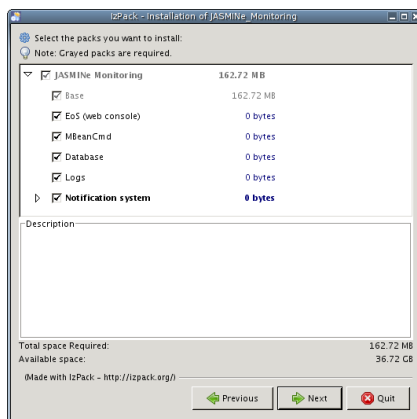
1. To execute the jar file, double-click on it or launch the command :

```
> java -jar jasmine-monitoring-installer-full.jar
```




When using the online installer, the installer is launched automatically after download.

2. After choosing the installation directory, accepting the license agreement, setting the JDK/JRE path, you can choose the components you want to install.



- *Core* contains JASMINe Monitoring core. It's required.
 - *EoS* is needed to install the web console.
 - *Probes* is used for collecting monitoring data.
 - *Database* is needed to store monitoring data in a database.
 - *Logs* is used to store monitoring data in log file(s).
 - *Notification system* is needed to create and manage notifications, alarms and automated actions.
3. Once you selected components, you can configure some ports used by JASMINe Monitoring.

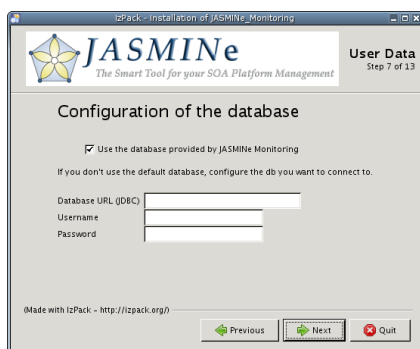


- *HTTP* port is the port on which the HTTP server of JASMINe Monitoring is listening to.
- *MBeanCmd* port is the port used to receive monitoring data from the running probes.

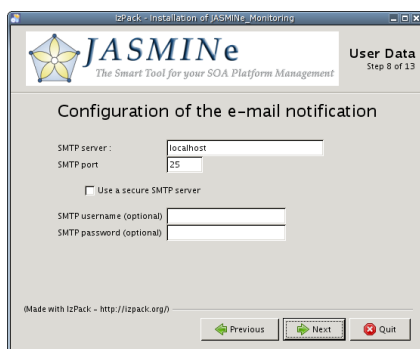
- *RMI port* is the registry port.
 - *JMS port* is used by the JMS server.
4. If you selected the component *Database*, you can specify the database you want to use to store data.

The default configuration uses HSQL database embedded in JASMINe Monitoring. This database can be used for tests but should not be used in a production environment.

To specify an other database, fill the URL of your database and authentication information.



5. If you selected the *E-mail notification* in the *Notification system* component, you have to specify a mail server (SMTP).




6. Once this is done, the installation can complete.



2.2.3. Start/Stop JASMINe Monitoring


- **Start**

JASMINe Monitoring can be started using the menu entry  *Start JASMINe Monitoring*.

You can also start JASMINe Monitoring in command-line :

```
> cd JASMINe_Monitoring
> jasmine start
```


- **Stop**

JASMINe Monitoring can be stopped using the menu entry  *Stop JASMINe Monitoring*.

You can also stop JASMINe Monitoring in command-line :


```
> cd JASMINe_Monitoring
> jasmine stop
```

- **Open the web console**

The menu entry  *JASMINe EoS (web console)* opens a web browser at the URL `http://localhost:9100/jasmine/`.

Use *jonas* for both user and password.

2.2.4. Uninstall JASMINe Monitoring

JASMINe Monitoring can be uninstalled using the menu entry  *Uninstall JASMINe Monitoring*.

You can also uninstall JASMINe Monitoring in command-line using the uninstaller jar file located in the directory `Uninstaller` :

```
> cd JASMINe_Monitoring/Uninstaller
> java -jar Uninstaller.jar
```

2.2.5. Manual installation

See Section 4.1, “Manual installation”.

2.3. EoS (Eye of SOA) - The web console

You can run the web console EoS by typing `http://localhost:9100/jasmine/` in your browser. If you specified a different port, use that instead of 9100.

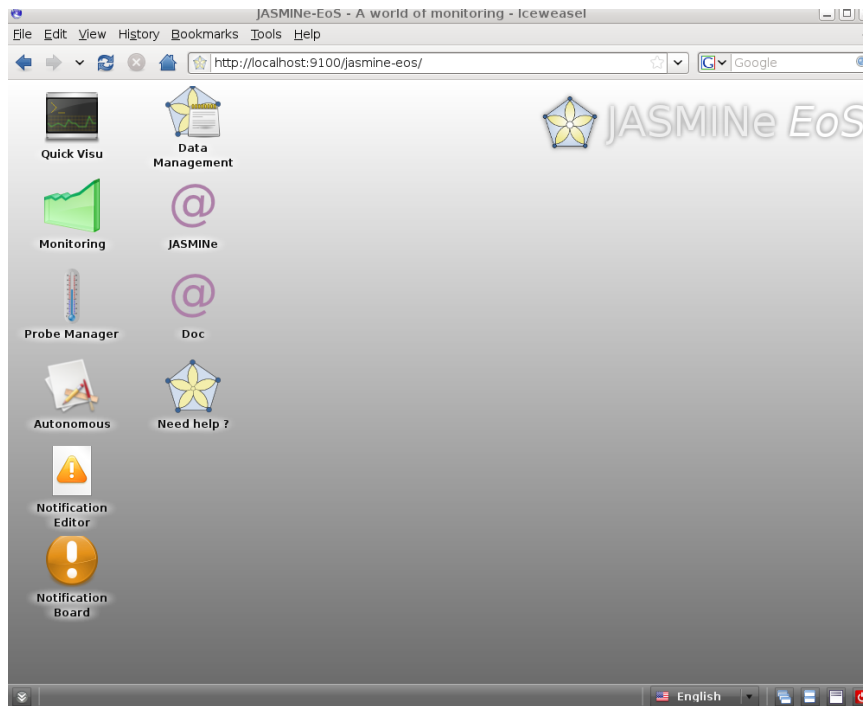
Log in using *jonas* for both username and password. To change the password, see Section 4.2.1, “Managing users and role”

2.3.1. Console features

The console is presented as a web desktop, where modules take place in separate windows. A taskbar provides the following controls:

- Minimize all windows
- Cascade windows
- Tile windows
- Show notifications log window

Figure 2.1. JASMINe EoS home page



- Use the *Probe Manager* module to collect monitoring data.
- *Quick Visu* and *Monitoring* modules provide graphs for real-time monitoring or post-mortem analysis.
- *Data Management* allows for management of the collected monitoring data.
- *Notification Editor* and *Notification Board* modules allow for creating notification rules and visualizing notifications corresponding to events in the monitored targets.



Tip

Switch to full-screen mode (usually F11) to enjoy JASMINe-EoS as a mini-OS !

Chapter 3. Working with JASMINe Monitoring

3.1. Collect monitoring data

Monitoring data is collected from target servers by JMX probes.

In order to start probes, MBeanCmd commands need to be executed.

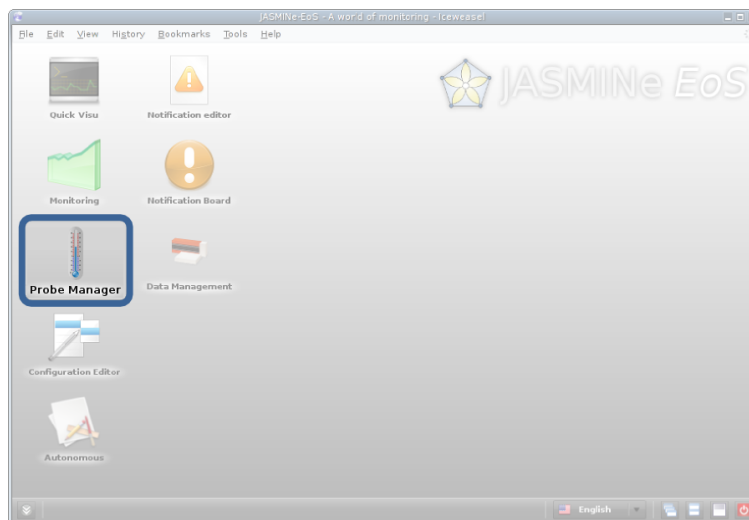
MBeanCmd commands can be executed in stand-alone mode (see the MBeanCmd documentation [http://jasmine.ow2.org/doc/trunk/doc-en/html/mbeancmd_guide.html] for more details) or via the Probe Manager.

The collected data may be sent to several destinations called outputs.

3.1.1. Collecting data with Probe Manager

Open the *Probe Manager* module

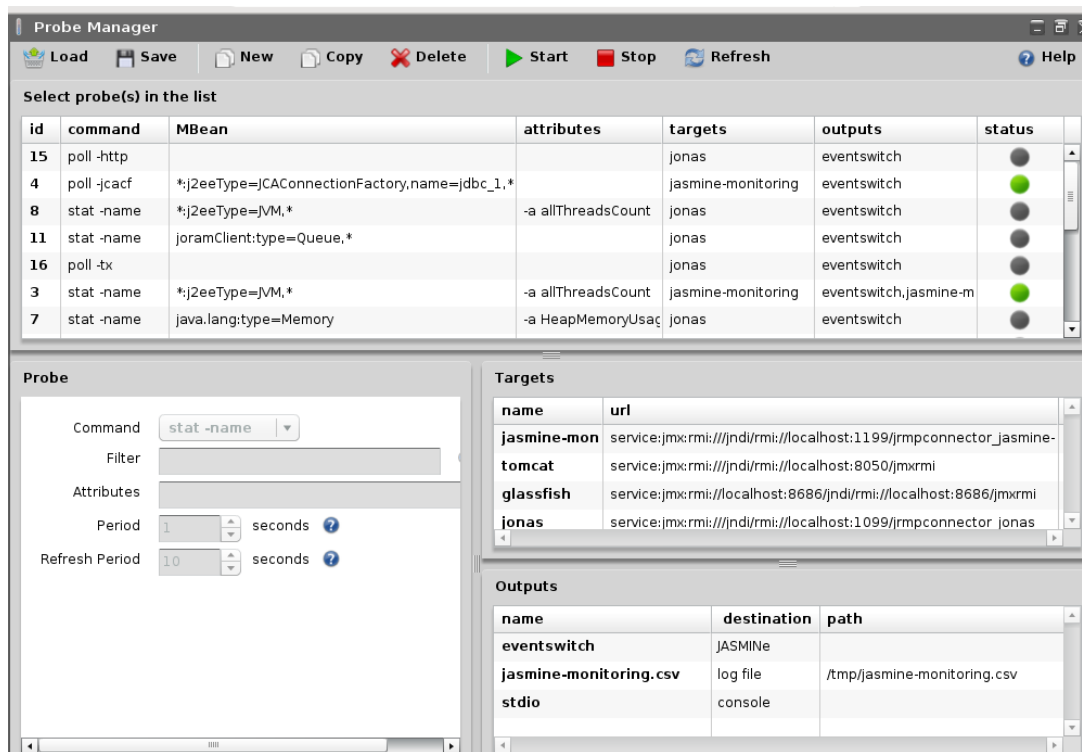
Figure 3.1. JASMINe EoS Console with probe manager icon



This module allows you to manage JMX *probes*, management *targets* and *outputs*, destinations for management data.

- You can list the existing probes, targets and destinations, create new ones, remove them, or modify them.
- You can visualize probe and target status, and change probe status by *start* or *stop* operations.
- All the probe, target and output definitions, together with probe status, can be *saved* in a configuration file.

When opening the *Probe Manager* module, the configuration file is *loaded*, and probes started (if their saved status is *started*).

Figure 3.2. Probe manager overview

On the top, a panel of commands is available :

- **Load** : allows to load a configuration file containing probe, target and output definitions. This file does not replace the existing probe definitions, but is merged with them.
- **Save** : save the current configuration to the "probe-config.xml" file. The probe status will be saved also, so that all the currently running probes will be launched when JASMINe Monitoring will be restarted.
- **New** : create a new probe, target or output definition.
- **Copy** : allows for creating a new probe by duplicating an existent probe definition.
- **Delete** : remove all the selected probes
- **Start** : start all the selected probes. If one of them was already started, no error is returned.
- **Stop** : stop all the selected probes. If one of them was already stopped, no error is returned.
- **Refresh** : refresh all the probes available in JASMINe Monitoring.
- **Help** : helps you if you need any indication

The layout below presents the list of available probes. The probes can be ordered by their properties (id, command, etc.). The status of each probe is indicated by a color :

- **Green** : the probe is running well
- **Orange** : the probe is starting
- **Red** : the probe is not running correctly
- **Black** : the probe is stopped

3.1.2. Managing probes

You can add a new probe by clicking on the *New* icon and selecting *New Probe*.

If you have to create a probe that uses a new target and/or output, start by creating them. Then, when creating the new probe, the already created target and output definitions will be proposed for choose in the popup. Currently, only one target and one output may be selected at creation time. If you need more than one target or output, you will have to first create the probe with only one, and then modify the probe to add more targets or outputs. This restriction will be removed in a next version.

For the rest of the new probe's parameters, the proposed values are those of the probe that was selected when starting the new operation.

The popup window allows you to select a MBeanCmd command. Currently available commands are *stat* and *poll*.

The *stat* command allows to collect data from MBeans attributes. The *Filter* parameter allows to select one or more MBeans, and the *Attributes* parameter allows to define the attributes to poll from the selected MBeans.

The *poll* command allows to collect specific monitoring data from a Java EE Application Server such as JOnAS. The different poll commands (tx, http, etc.) use predefined MBean filters that match the specific resources in the server. For example, *poll tx* uses standard J2EE MBeans that provide statistics about transactions in the server (number of committed, rolledBack or timedOut transactions, etc.)

Figure 3.3. Probe creation popup

The screenshot shows a 'Create a new Probe' dialog box. It contains the following fields and controls:

- command ***: A dropdown menu with 'stat -name' selected.
- MBean**: An empty text input field.
- attributes**: An empty text input field.
- target ***: A dropdown menu with 'jasmine-monitoring' selected.
- output ***: A dropdown menu with 'eventswitch' selected.
- polling period (sec.)**: A spinner box with the value '10'.
- refresh period (sec.)**: A spinner box with the value '300'.
- Buttons**: 'Submit' and 'Cancel' buttons at the bottom.

The other way to create a probe is to duplicate an existing one using the *Copy* operation. You can then modify the initial definition. For example, you can add one or more targets, or remove targets.

By clicking on a probe that is stopped, you will see that all the bottom part is used to display the properties of that probe. On the left side, you will be able to change the command (stat or poll), the filter, the arguments and the polling period and the MBeans' refresh period. If you click on *check filter*, you will see all the MBeans corresponding to the filter and available on the selected target.

Figure 3.4. Probe detail in edition

On the the right side of the window, you can select on or more targets by clicking on their names. Press *Ctrl* on your keyboard to select more than one. You can do the same for the outputs on the bottom area. Click on *Apply changes* button to validate your modification or *Cancel* to abort them. When you finished editing, you can start the probe.

If you start a probe having a *jasmine* output, you can see the collected data obtained by the probe at the left of the bottom area.

Figure 3.5. Probe detail

time	attribute	value
13:39:59 Gi	currentCpuLoad	5.7
13:39:59 Gi	currentCpuTime	570
13:39:59 Gi	processCpuTime	33650
13:39:54 Gi	currentCpuLoad	0
13:39:54 Gi	currentCpuTime	0
13:39:54 Gi	processCpuTime	33080

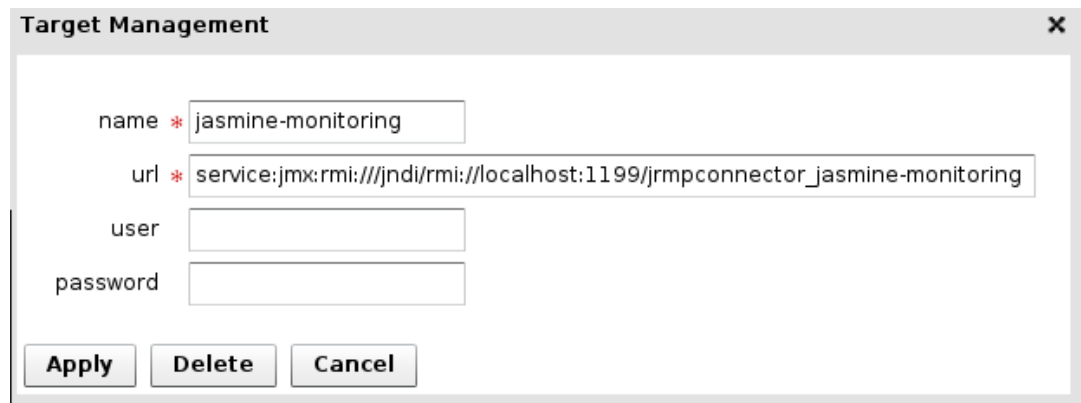
3.1.3. Managing targets

By clicking the *New* icon, you will be able to add new a target. The target management popup will appear, requesting the *name* and the JMX *url* necessary to identify the target, and to allow connection to be established when starting probes. You can specify a user *login* and *password* if needed.

Target naming policy is left to the JASMINe Monitoring user, as it often depends on particular administration rules. The only constraint is to have unique names for targets. Target names will appear as metadata inside the data flow returned by running probes.

The same popup will appear if you click on the *edit* icon near the target. You will be able to change the properties or to delete the target. You can also click on the *magnifying glass* to see the list of all the MBeans deployed on the target. The colored circle in front of each target gives its status. If it is *green* it means that there are some probes running well on the target. If it is *orange* it means that there are starting probes. If there are some problems it will be *red* and if there are no probes running, it will be *black*.

Figure 3.6. Target Management

A screenshot of a 'Target Management' dialog box. The dialog has a title bar with 'Target Management' and a close button (X). Inside, there are four labeled text input fields: 'name' with the value 'jasmine-monitoring', 'url' with the value 'service:jmx:rmi:///jndi/rmi://localhost:1199/jrmpconnector_jasmine-monitoring', 'user' (empty), and 'password' (empty). At the bottom, there are three buttons: 'Apply', 'Delete', and 'Cancel'.

3.1.4. Managing outputs

You can add a new output one by clicking the *New* icon and selecting *New Output*. Then, an output management popup will appear, giving you the possibility to define an output *name* and select a *destination*. The proposed destinations are:

- *console* allowing to see monitored data in the console where JASMINe Monitoring was started.
- *logfile* allowing to save monitored data in a file.
- *jasmine* allowing to make data available for the other JASMINe EoS modules, like *Quick Visu* or *Monitoring*. Data collected is kept in the jasmine monitoring database, and will be available later for monitoring replay operation.

The same popup will appear if you click the *edit* icon in front of an output. You will be able to change its properties and to delete it.

Figure 3.7. Output Management

Output Formats

The monitoring data format produced by probes depends on the destination type.

For *console* and *logfile* a header is produced at the first polling, then, the requested indicators' values are provided together with meta information related to polling date and time, target identification, and the name of the MBeans providing the data.

Here is a header for the following probe:

```
stat -name java.lang:type=GarbageCollector,* -a CollectionTime -target tomcat
```

```
date;time;sname;server;domain;mbean;cmdid;CollectionTime
```

And an example of output sequence

```
2011/04/14 13:30:31;1302784231085;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS Scavenge;3;165
2011/04/14 13:30:31;1302784231087;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS MarkSweep;3;2848
2011/04/14 13:30:36;1302784236088;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS Scavenge;3;165
2011/04/14 13:30:36;1302784236089;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS MarkSweep;3;2848
2011/04/14 13:30:41;1302784241090;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS Scavenge;3;165
2011/04/14 13:30:41;1302784241091;service:jmx:rmi:///jndi/rmi://localhost:12345/
jmxrmi;tomcat;unknown_domain;java.lang:type=GarbageCollector,name=PS MarkSweep;3;2848
```

3.1.5. Start collects at JASMINe Monitoring startup

Collects can be started at JASMINe Monitoring startup. To do that, some probes (MBeanCmd commands) must be defined in the probes configuration file `conf/probe-config.xml`, where `conf` is located in the JASMINe Monitoring's installation directory. The probes to be started must have status equal to *started*.

The probes configuration file must contain the definition of the targets and outputs used by the probes to be started.

```
<probe-config xmlns="org.ow2.jasmine.monitoring.mbeancmd:probe-config">
  <target url="service:jmx:rmi://localhost:8686/jndi/rmi://localhost:8686/jmxrmi"
  id="glassfish"/>
  <target url="service:jmx:rmi:///jndi/rmi://localhost:1099/jrmpconnector_jonas"
  id="jonas"/>
  <target url="service:jmx:rmi:///jndi/rmi://localhost:1199/jrmpconnector_jasmine-
  monitoring" id="myself"/>
```

```

<target url="service:jmx:rmi:///jndi/rmi://localhost:12345/jmxrmi" id="tomcat"/>

<output host="localhost" dest="jasmine" id="eventswitch"/>
<output path="/tmp/jasmine-monitoring.csv" dest="logfile" id="jasmine-monitoring.csv"/>
<output dest="console" id="stdio"/>

<probe status="started" args="-a CollectionCount CollectionTime" separator=";"
refreshPeriod="900" period="5" filter="java.lang:type=GarbageCollector,*" cmd="stat">
  <target>tomcat</target>
  <output>stdio</output>
</probe>

<probe status="stopped" args="-a HeapMemoryUsage NonHeapMemoryUsage " separator=";"
refreshPeriod="900" period="5" filter="java.lang:type=Memory" cmd="stat">
  <target>myself</target>
  <output>stdio</output>
</probe>

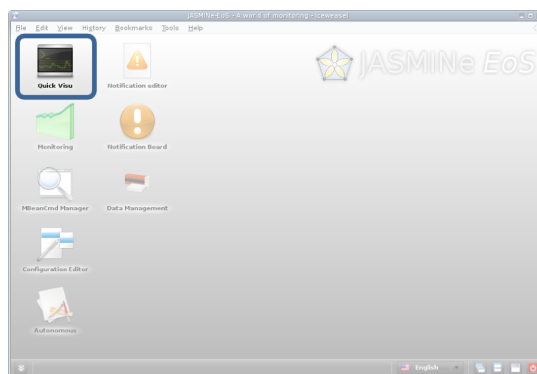
<probe status="stopped" separator=";" refreshPeriod="900" period="10"
filter="*:j2eeType=JCAConnectionFactory,name=jdbc_1,*" which="jcacf" cmd="poll">
  <target>jonas</target>
  <output>eventswitch</output>
</probe>
</probe-config>

```

3.2. Monitoring

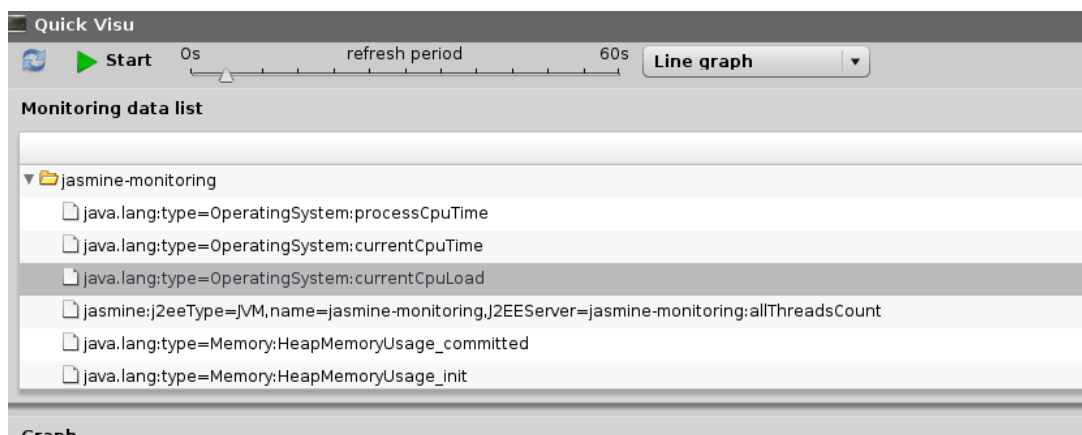
3.2.1. Simple monitoring

To quickly get monitoring charts, use the *Quick Visu* module.



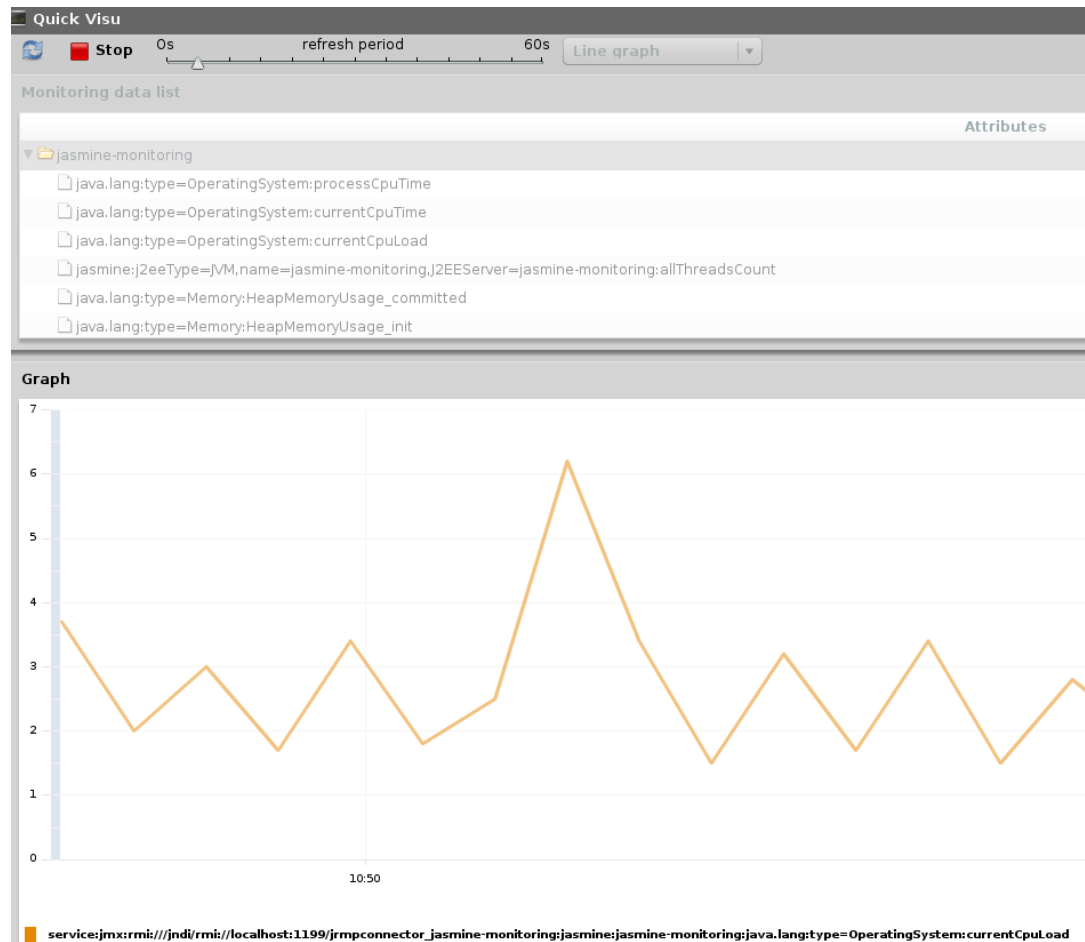
Once you opened the module, all the available monitoring data is displayed in a table.

These data correspond to indicators returned by the running probes. Indicators are grouped using the name of polled target. The target url can be seen by placing the mouse on any element in the list.



Select indicators you want to display in a chart, then click *Start* button.

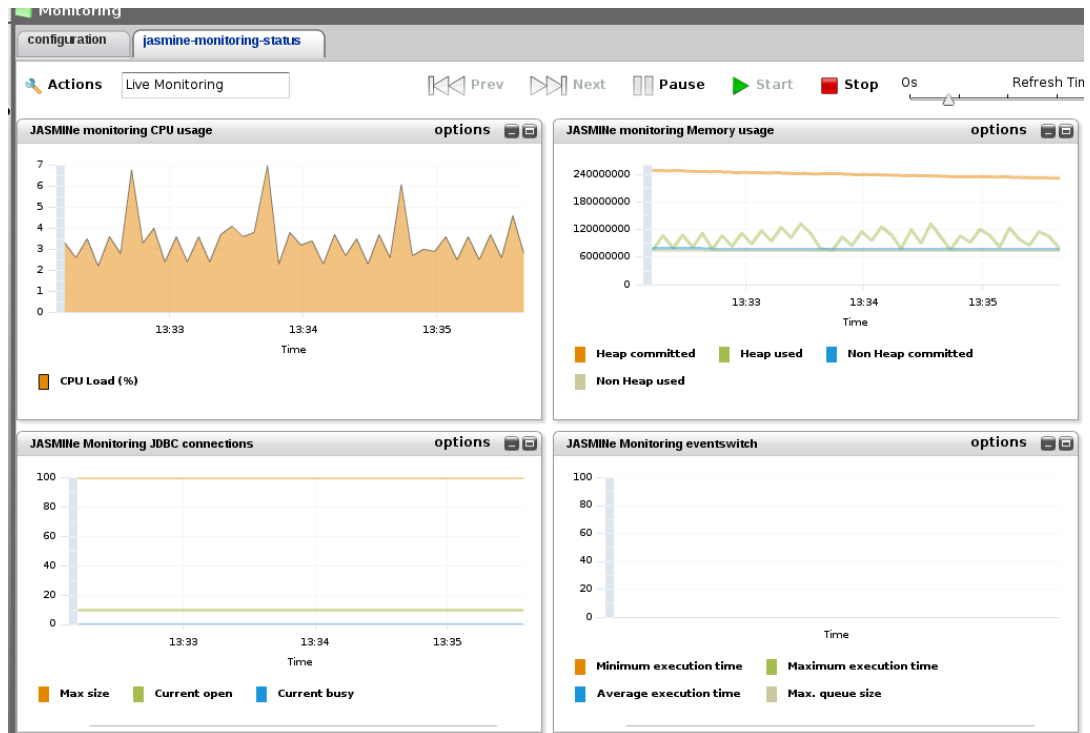
If you want to monitor other indicators corresponding to a new started probe, stop the current monitoring with the *Stop* button, and refresh the monitored data list with a click to the *Refresh* button.



3.2.2. Advanced monitoring

Monitoring is an advanced visualization module, which allows to monitor data

- in real time,
- from a database,
- from a local CSV log file, or
- from CSV log file located on the JASMINe server.



Monitoring dashboards can be customized by defining graphs with one or more series of indicators corresponding to data you want to monitor. The configuration of a dashboard is stored in a XML file.

The list of available configuration files is presented to the user when opening the module. These files are stored in the `monitoring` directory under the JASMINe Monitoring installation directory.

You have to select a dashboard configuration file in order to create a new dashboard and to do monitoring.

You may also import new configuration files, modify existing files, rename and remove configuration files.

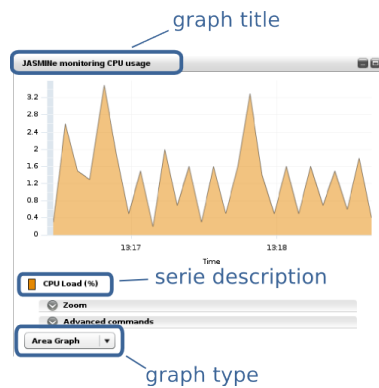
3.2.2.1. Create a new dashboard

A dashboard is composed of graph(s) which are composed of series.

A graph has a title and is defined by its type (line, area, column).

A series is characterized by an id, a name and a description. The 3 most important attributes of a series are the **indicator name**, the source **server** and the **pattern** filter.

- The indicator name, defined using the **y** attribute, must matches the name of a collected data (indicator name).
- The server source (optional) is used to filter indicators on the monitored target.
- The pattern (optional) is a JMX MBean name pattern that can be used to filter the monitoring data returned by probes.



Examples of such files are available in the directory `monitoring` on the JASMINe Monitoring server.

Here is an example of file :

```
<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE monitor PUBLIC
  "-//OW2//DTD JASMINe MBeanCmd 1.0//EN"
  "http://jasmine.ow2.org/dtds/graph.dtd" >
<monitor absciss="time" separator=";" timeformat="long">

<serie-def id="currentCpuLoad">
<name>Current CPU Load</name>
<description>CPU Load (%)</description>
<y col="java.lang:type=OperatingSystem:currentCpuLoad" type="double" />
<server>jasmine-monitoring</server>
</serie-def>

<graph id="cpu">
<title>JASMINe monitoring CPU usage</title>
<serie ref="currentCpuLoad" />
<chartType>areaChartConst</chartType>
</graph>

</monitor>
```

3.2.2.2. Live monitoring

This is the default operating mode.

You can start monitoring with **Start** button. The graphs are automatically refreshed using a refresh period that can be dynamically modified.

If you want to control manually graphs refreshing, use **Next** button.

3.2.2.3. Replay from server file

Change mode using *replay server file* in **Actions**. A list of files on the server is proposed for choose. By default, `.csv` log files are listed. Such files can be generated by probes having a *file* output.

3.2.2.4. Replay from local file

Change mode using in **Actions** *replay local file*.

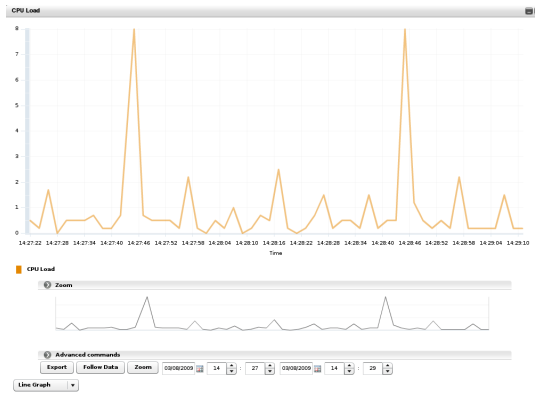
3.2.2.5. Replay from database

Change mode using in **Actions** *replay database*.

3.2.3. Chart navigation

You can navigate in the displayed data by using the graph zoom. This can be done globally, using the **zoom** option in **Actions**, or, for a particular graph, using that graph's options.

You can also follow the last arrived data by activating *follow* option in **Actions**. It will keep your window opened on the data you are looking at, and update the data when new one is arriving.



You can export the graph you got as a .png file by using on the export graph option. It will open a new tab in your navigator with the picture, you will just have to save it.

3.2.4. Post-mortem analysis

The *Monitoring* module offers a *Replay* functionality.

Once your configuration is loaded, you have 2 possibilities :

- Replay data from a log file.

Change mode using in **Actions** *replay server file*, or *replay local file*.

- Replay data from the JASMINe Monitoring server Database.

Change mode using in **Actions** *replay database*.

Then define replay period using the pop-up window.

Database :

08/01/2010	10	:	30
09/01/2010	22	:	30

3.3. Management of Monitoring Data

3.3.1. Using the Web Console

The *Data Management* module allows you to manage data stored in the JASMINe server's database.

Currently, you can do a purge of the database following various criteria like record number, record age, monitoring data of a specific server and/or a specific domain.

You also can register purge tasks which can be executed periodically.

3.3.2. Purge Command

The above operations are also accessible in a command line tool.

To launch a purge, you have to use the `purge.jar` by launching :

```
java -jar purge.jar purge -maxAge | -maxNumber [-startDate -execPeriod] [-serverName -
domainName]
```

If no option are specified, this will simply display help on the available options.

Options that allow to specify purge criteria are: **maxAge** and **maxNumber**. You have to choose one of them.

By default, the purge operation is done on all the monitoring data regardless to the target server. This can be changed using the **serverName** and **domainName** options

Purge tasks can be defined using **startDate** and **execPeriod** options.

By default, execution is launched immediately.

- `-maxAge daysDhoursHminutesM` : Entries older then `maxAge` will be purged
- `-maxNumber recordsNumber` : Number of records that are preserved
- `-serverName name` : Name of the concerned managed server
- `-domainName name` : Name of the concerned management domain
- `-startDate yyyy/MM/dd-HH:mm` (example: 2010/10/15-10:30) : Date to trigger purge task. If not defined, current date is used for starting a periodic execution.
- `-execPeriod daysDhoursHminutesM` (example: 1H10M means every 1 our and 10 minutes) : Execution period

Examples

- `purge -maxAge 1h`

Purge all data older then 1 hour

- `purge -maxAge 1d -execPeriod 1d`

Periodically purge all data older then 1 day, starting with this moment



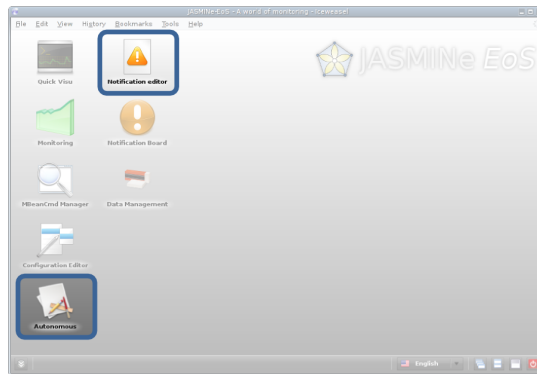
Note

The Purge command is using an EJB deployed in the JOnAS Server hosting JASMINe. Its JNDI name is "db-ejb/timer". Connecting to the EJB requires the **smarclient** service to be activated.

3.4. Notification

3.4.1. Autonomous & Notification editor

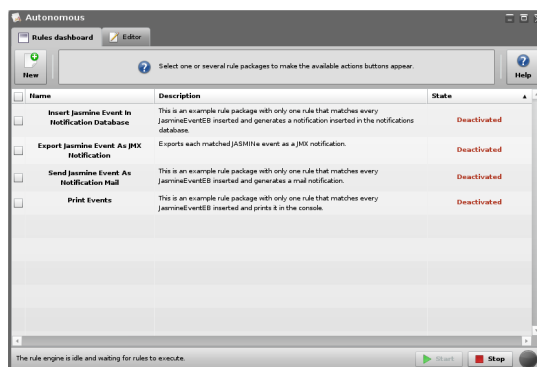
The *Autonomous* and *Notification Editor* modules allow to create, edit and manage rules. *Autonomous* manages rules that trigger autonomic actions on the system (restart server, etc). *Notification Editor* manages rules that generate notifications.



Rules will be executed in rule engine(s) (JASMINe Monitoring uses Drools 5.x [<http://www.jboss.org/drools/>]). *Autonomous* and *Notification Editor* manage different instances of the rule engine, identified by their names. So that, everything you do in one of these modules cannot affect the other.

Those two modules are very similar, and composed of three components :

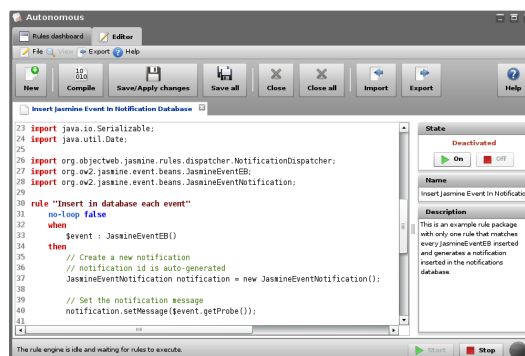
- The rule editor to create and edit rules.
- The rule engine dashboard which shows the state of the rule engine (idle, started, or stopped).
- The rules manager which allow you to manage, activate and deactivate rules.



3.4.1.1. The rule editor

Two edition modes are available:

- **Visual mode:** coming soon
- **Textual mode:** edit the source of the rule directly.





Note

The rules are written in Drools language (See Drools documentation [<http://downloads.jboss.com/drools/docs/5.0.1.26597.FINAL/drools-expert/html/ch04.html>]).

The rule editor allows to edit multiple rule packages at the same time in tabs.

The rule editor and rule manager will not let you activate rule packages if they don't compile. Correct the errors before activation, or save it deactivated.

The rules editor and rule manager editor can export rule packages as files, in two output formats:

- As source file: generates an archive containing the Drools source files (.drl) of the selected rules.
- As bundle: Generate an OSGi bundle that can be directly deployed to activate the embedded rule packages. This allows to generate "ready-to-deploy" profiles and not use JASMINe-EoS to manage your rules. Be conscious that rules activated this way will not appear in the list of the managed rules packages.

3.4.1.2. The rule engine dashboard

The rule engine dashboard is the status bar of the module. You can monitor in real-time the status of the managed rule engine instance, start it and stop it.

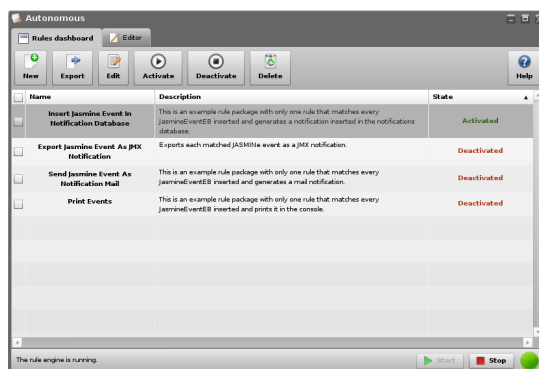


If the configuration is incorrect and the rule engine service is unreachable, it displays a "connexion issue" symbol. The rule engine statuses can be:

- **Idle (grey light):** This is the default state at startup. It means the rule engine is ready and waiting for rules to be executed. As soon as a rule package is activated successfully, it switches to "Running" state.
- **Running (green light):** When rule packages are activated and running.
- **Stopped (red light):** When the rule engine has been stopped manually. In this state, even if you activate rule packages, the rule engine will stay in standby and wait for the "start" order to execute them. This is a safe state to be on when modifying active rules...

3.4.1.3. The rule manager

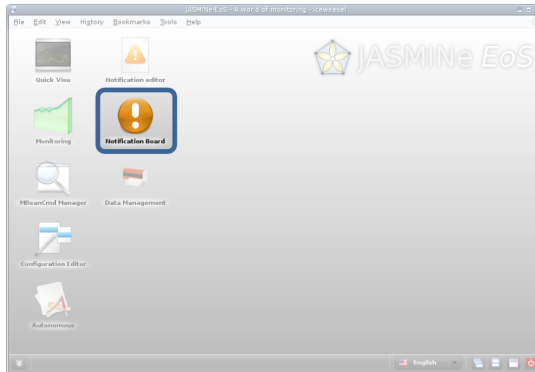
In this list, you can manage, activate, deactivate and delete the rules on the rule engine. This can be done individually for each rule package, by clicking on the status cell. You can also activate/deactivate/delete several rule packages in one operation by selecting multiples rows.



Every action performed on rule packages must be confirmed before being effective: modification, activation and deactivation, to avoid bad surprises and unwanted behaviours of the rule engine. This is why when you click on "activate" on a rule package, this is marked as "activation requested". The activation is effective only when you click on "Apply changes".

3.4.2. Notification board

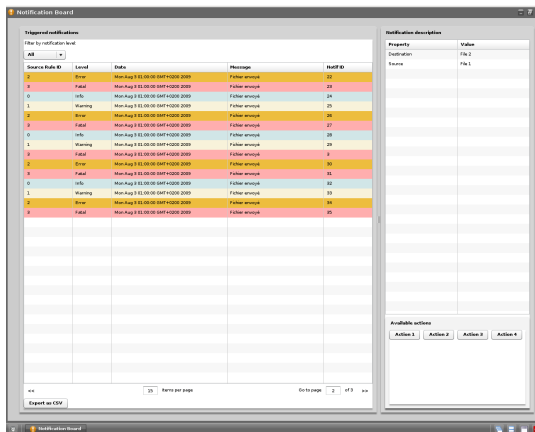
Notification board module displays the notifications generated by JASMINe Monitoring.



3.4.2.1. Displaying data

The notification board is composed of two parts :

- A table containing the notifications.
- A panel displaying details and manual actions available for the selected notification.



The table contains the notification attributes :

- The id of the notification.
- The level.
- The time and date of the notification.
- The message associated with the notification.
- The id of the rule that generate the notification.

When a notification is selected, its specific properties are shown in the "Notification description" table. This table presents properties in a key-value manner.

The module allows you the export of the notification table as a CSV file. For that, press the "Export as CSV" button below the table. The used separator is ; (semicolon).

3.4.2.2. Interface configuration

The configuration of the Notification board interface is made by a configuration file, called `notification-config.xml`. This file can be found in the `META-INF` folder of the `jasmine-eos.war` archive.

The file has several sections:

- The `levels` section defines the levels of the notifications and the colors associated with each level.
- The `acknowledgeClass` tag represents the Java class that is called when a notification is acknowledged.
- The `notifs` section contains all the notifications that will be displayed on the Notification board. For each `<notif>` tag, the attributes are:
 - the `rule` attribute that represents the id of the rule that created the notification. If this attribute is omitted, the notification won't be displayed on the board.
 - the `level` attribute that represents the level of gravity of the notification. If a color is defined for this level, the notification will also be associated with this color. What happens if the level is not defined in the `<levels>` section?
 - the `synchronized` attribute

Each notification can be associated with actions. The `<actions>` tag presents these actions and the java classes that are called when an action is requested.

Chapter 4. Advanced features

4.1. Manual installation

4.1.1. Requisites

- JOnAS v5.2.x [<http://jonas.ow2.org/>]
- Apache ANT (≥ 1.7) [<http://ant.apache.org/>]
- Java JDK (≥ 1.5) [<http://www.oracle.com/technetwork/java/javase/downloads/index.html>]

4.1.2. Configuration

The manual installer of JASMINe Monitoring is based on Apache ANT. A set of tasks are defined to install and configure JASMINe components.

A properties file, `jasmine_installation.properties`, centralized the configuration.

- `INSTALL_PATH` : Path of the directory where JASMINe Monitoring will be created.
- `JONAS_ROOT` : Path of the JOnAS application server directory.
- `JAVA_HOME` : Path of the Java JDK installation.

The list of JASMINe Monitoring components to install. Values must be `true` or `false` :

- `web-console` : Install the web console JASMINe EoS.
- `probes` : Is used for collecting monitoring datas.
- `database` : Is needed to store monitoring datas in a database.
- `notification` : Is needed to create and manage notifications and alarms.
- `selfmanagement` : s needed to create and manage automated actions.
- `log` : Stores monitoring datas in log file(s).

List of main ports used :

- `http.port` : Port is the port the HTTP server of JASMINe Monitoring is listening to.
- `mbeancmd.port` : Port used to receive monitoring datas.
- `rmi.port` : Port of the registry.
- `jms.port` : Port for the JMS server.

Database configuration :

- `database.type` : Currently could be `hsq1`, `mysql`, `oracle` or `postgresql`.
- `database.name` : Name of the database.

- `database.host` : Host of the database.
- `database.port` : Port of the database.
- `database.username` : User of the database.
- `database.password` : Password for the previous user.
- `database.jdbc.driver.path` : Path to a JDBC driver which will be copied to the JASMINe Monitoring server to be used.

SMTP (email server) configuration :

- `smtp.server` : Host of the SMTP server.
- `smtp.port` : Port of the SMTP server.
- `smtp.need.authentication` : Is the SMTP server secured. Values must be `true` or `false`.
- `smtp.username` : If the server is secured, name of the user.
- `smtp.password` : If the server is secured, password of the user.

Registration :

- `user_agree` : Do you want to register JASMINe Monitoring ? Value must be `true` or `false`.
- `user.email.address` : Email used for the registration. Could be blank
- `installer.proxyaddress` : If your internet connection is under an HTTP proxy, host of the proxy.
- `installer.proxyport` : Port of the proxy.
- `installer.proxyuser` : If the HTTP proxy is secured, name of the user.
- `installer.proxypwd` : If the HTTP proxy is secured, password of the user.

4.1.3. Installing JASMINe Monitoring

Once you have configured the JASMINe Monitoring installer by editing `jasmine_installation.properties`, you just have to launch

```
ant -f jasmine_installation.xml
```

4.1.4. Starting JASMINe Monitoring

See Section 2.2.3, "Start/Stop JASMINe Monitoring"

4.2. Administrate JASMINe Monitoring

4.2.1. Managing users and role

To add or modify a JASMINe monitoring user, you have to edit the file `$JASMINe_MONITORING/conf/jonas-realm.xml`. Add an entry to the `<users>` element with a *name*, a *password* (which can be encrypted or not) and either a *groups*="*jonas*" or *role*="*tomcat*".

The default user for JASMINe EoS is *jonas*. To modify the default password (*jonas*), change it in the previous configuration file.