Mobicents JAIN SLEE
Jopr Plugin User Guide

by Alexandre Mendonça, Douglas Silas, Ivelin Ivanov, and Jared Morgan

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts* [https://fedorahosted.org/liberation-fonts/] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight key caps and key-combinations. For example:

> To see the contents of the file `my_next_bestselling_novel` in your current working directory, enter the `cat my_next_bestselling_novel` command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

> Press **Enter** to execute the command.
>
> Press **Ctrl**+**Alt**+**F1** to switch to the first virtual terminal. Press **Ctrl**+**Alt**+**F7** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in `Mono-spaced Bold`. For example:

> File-related classes include `filesystem` for file systems, `file` for files, and `dir` for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

`Mono-spaced Bold Italic` or ***`Proportional Bold Italic`***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type `ssh` *`username@domain.name`* at a shell prompt. If the remote machine is `example.com` and your username on that machine is john, type `ssh john@example.com`.
>
> The `mount -o remount` *`file-system`* command remounts the named file system. For example, to remount the `/home` file system, the command is `mount -o remount /home`.
>
> To see the version of a currently installed package, use the `rpm -q` *`package`* command. It will return a result as follows: *`package-version-release`*.

Note the words in bold italics above username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as

a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventions

Two, commonly multi-line, data types are set off visually from the surrounding text.

Output sent to a terminal is set in `Mono-spaced Roman` and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in `Mono-spaced Roman` but are presented and highlighted as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }

}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss.

2. Provide feedback to the authors!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: *http://bugzilla.redhat.com/bugzilla/* against the product **${product.name}**, or contact the authors.

When submitting a bug report, be sure to mention the manual's identifier:

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Introduction to Jopr and Mobicents JAIN SLEE Plugin

Jopr is the JBoss/Red Hat solution for enterprise management which provides administration, monitoring, alerting, operational control and configuration in an enterprise setting with fine-grained security and an advanced extension model.

More information regarding Jopr can be found at the official page [http://www.jboss.org/jopr]

Mobicents JAIN SLEE takes advantage of this full-featured platform, with its custom plugin. This allows complete management, such as deployment and installation of JAIN SLEE Deployable Units, creation and activation of Resource Adaptor Entities, etc. as well as monitoring critical resources, such as the number of Activities present, SBB Entities, etc.

Since Jopr relies on a plugin system, the same instance can be used to administer/monitor Mobicents JAIN SLEE, Mobicents SIP Servlets and Mobicents Media Server, providing a seamless environment for managing the complete suite, including the hosting JBoss AS instances as well.

Installing Jopr and Mobicents JAIN SLEE Plugin

2.1. Pre-Install Requirements and Prerequisites

Ensure that the following requirements have been met before continuing with the install.

2.1.1. Hardware Requirements

Sufficient Disk Space

Once unzipped, the Jopr binary release requires *at least* 300MB of free disk space. Keep in mind that disk space requirements may change from release to release.

Anything Java Itself Will Run On

The Jopr platform will run on the same hardware that the JBoss Application Server runs on, but it is recommended at least 2GB or 4GB of RAM memory, for 32 or 64 bit Operating Systems.

2.1.2. Software Prerequisites

JDK 5 or Higher

A working installation of the Java Development Kit (JDK) version 5 or higher is required in order to run the Jopr. Note that JBoss Application Server is a runtime dependency, but comes bundled with the binary distribution.

For instructions on how to install the JDK, refer to *Appendix A, Java Development Kit (JDK): Installing, Configuring and Running*.

External Database

In order to run Jopr, an external database must be installed.

Use of Embedded H2 database

Jopr is distributed with a Java embedded database called "H2", so if you are just running Jopr to test it out and play with it, you can use that and not install an external database. But this Embedded H2 database is to be considered only for demo'ing and testing purposes. For production systems, you will need to install an external database.

The supported databases are:

PostgreSQL

Postgres versions 8.2.4 and higher are supported.

Oracle

Oracle versions 10g and 11g are supported.

After installing the database, make note of the JDBC URL, username and password for the database. This information is required during the Jopr Server installation.

2.2. Install Jopr and Mobicents JAIN SLEE Plugin

2.2.1. Installing Jopr Server and Agent

Jopr is based on and plugin-compatible with the multi-vendor RHQ management project. For the Mobicents Plugin to work correctly, version 1.4.x of RHQ is required.

Since there's no released version of Jopr based on RHQ 1.4.x yet, RHQ itself will be used instead.

1. Download RHQ Server from *SourceForge* [http://sourceforge.net/projects/rhq/]. This is the platform server, which will be responsible for storing, processing and presenting the data received by the agent(s).

2. Unzip the distribution in a suitable place (i.e. in a directory not too deeply nested and with no spaces or non-ASCII characters in its name)

3. *This step is optional, if you're using Embedded H2 Database, you can skip to next*

   Install a database like PostgreSQL (version 8.2.4 +, please choose C locale for initdb), create a database instance called 'rhq' in it and a user 'rhqadmin' that owns this 'rhq' database.

   ```
   create user rhqadmin password 'rhqadmin';

    create database rhq owner rhqadmin;
   ```

4. Eventually set RHQ_SERVER_JAVA_HOME or RHQ_SERVER_JAVA_EXE_FILE_PATH env variables appropriately. This is needed if your JAVA_HOME does not point to a java installation that is valid with Jopr (e.g. JAVA 1.4); Jopr requires JAVA 5+.

5. In a terminal console, cd into the installation directory and start the Server:

   ```
   bin/rhq-server.[sh|bat] console
   ```

6. After a few seconds, the messages on the console will stop. When this happens, point your browser to http://localhost:7080/ and run the installer:

   a. Click on **Click here to continue the installation**

   b. Choose your desired and installed database from the **Database Type** drop down list. If you don't want to install/use an external Database, choose **H2 (embedded)**.

   c. Default settings should be fine to install the server correctly. To test the DB connection, please click **Test Connection** button.

Warning

If the **Test Connection** button was used, make sure that the **Database Password** field is still filled, as some browsers lose the data in there. Default password is *rhqadmin*.

d. Review the seleceted settings, click **Install Server!** and wait for it to complete.

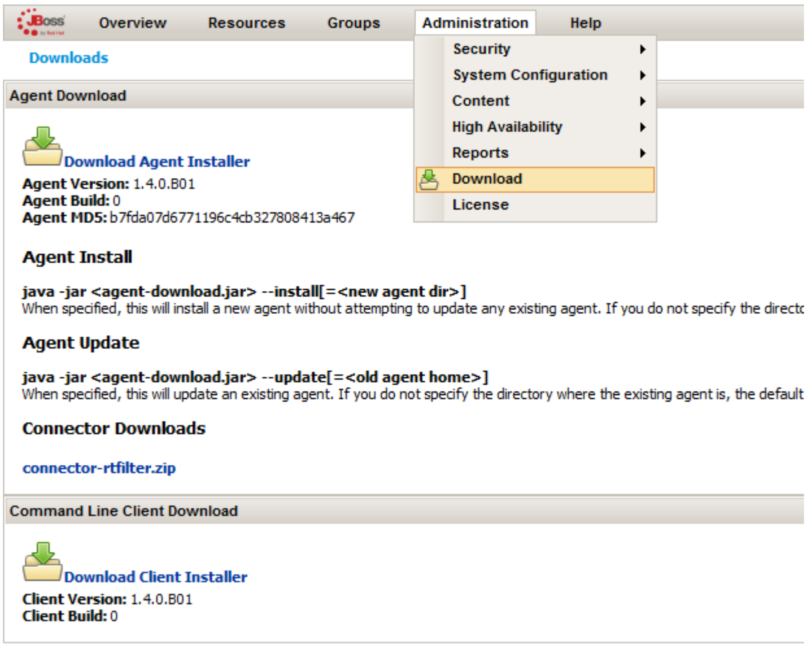
e. Click on **Done! Click here to get started!** when the progress bar stops.

7. Log in (default user/password are: rhqadmin/rhqadmin), it is time to download and install the Agent.

   The Agent is the component where our plugin will be running at. It is responsible for communicating with the managed resource (in this situation, the JAIN SLEE Server) to obtain vital data (eg: availability), metrics, operations execution, etc. and report back to the Server, who will process and display that data.

   To download it, follow these steps:

   a. Click on **Administration** → **Download**.



Agent Version: 1.4.0.B01
Agent Build: 0
Agent MD5: b7fda07d6771196c4cb327808413a467

Agent Install

```
java -jar <agent-download.jar> --install[=<new agent dir>]
```

When specified, this will install a new agent without attempting to update any existing agent. If you do not specify the directc

Agent Update

```
java -jar <agent-download.jar> --update[=<old agent home>]
```

When specified, this will update an existing agent. If you do not specify the directory where the existing agent is, the default

Client Version: 1.4.0.B01
Client Build: 0

Figure 2.1. Accessing Download Menu

   b. Click on **Download Agent Installer** and save it to where you wish to have it installed.

   c. In a second terminal console, cd into the directory containing the downloaded jar and install it:

```
java -jar <agent-download.jar> --install[=<new agent dir>]
```

The optional parameter, `new agent dir` indicates where the files will be installed, and defaults to the path where the jar is at.

d. cd into the newly created *rhq-agent* directory and start the Agent by issuing:

```
bin/rhq-agent.[sh|bat]
```

e. Answer the questions asked by the Agent, if any. Default setting should be sufficient.

f. There should be a console with a '> ' prompt. Don't close it, leave it running, it's the agent who will perform all the work.

For more detailed and up to date information, visit the Jopr *installation instructions* [http://jboss.org/community/docs/DOC-12828].

2.2.2. Installing the Mobicents JAIN SLEE Plugin

Mobicents JAIN SLEE Plugin is the interface between Jopr and Mobicents JAIN SLEE. It is responsible for gathering data from SLEE and transform it to Jopr format. It is an agent plugin, since it runs on the agent side.

Mobicents JAIN SLEE Plugin, may be installed in two ways:

Using the GUI

In the Jopr GUI, in the top menu, head to **Administration** → **System Configuration** → **Plugins**:

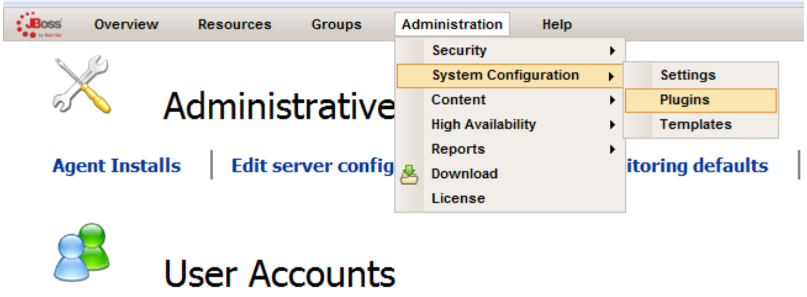
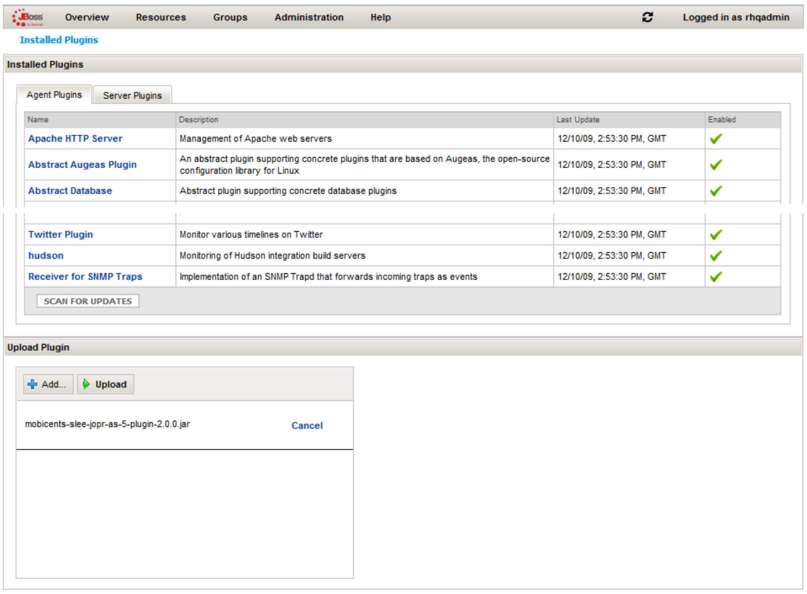


Figure 2.2. Accessing Plugins Menu

On the **Agent Plugins** tab (selected by default), on the bottom of the page, click **Add...**, browse to the Jopr plugin JAR file and select it. It will be listed for upload, as it can be seen in the following figure:



Name	Description	Last Update	Enabled
Apache HTTP Server	Management of Apache web servers	12/10/09, 2:53:30 PM, GMT	✓
Abstract Augeas Plugin	An abstract plugin supporting concrete plugins that are based on Augeas, the open-source configuration library for Linux	12/10/09, 2:53:30 PM, GMT	✓
Abstract Database	Abstract plugin supporting concrete database plugins	12/10/09, 2:53:30 PM, GMT	✓
Twitter Plugin	Monitor various timelines on Twitter	12/10/09, 2:53:30 PM, GMT	✓
hudson	Monitoring of Hudson integration build servers	12/10/09, 2:53:30 PM, GMT	✓
Receiver for SNMP Traps	Implementation of an SNMP Trapd that forwards incoming traps as events	12/10/09, 2:53:30 PM, GMT	✓

mobicents-slee-jopr-as-5-plugin-2.0.0.jar

Figure 2.3. Uploading Plugin

Finally, click the **Upload** button and the plugin will be deployed. Click **Scan for Updates** and it should be shown in the list as "Mobicents JSLEE Server 2.x".

If the agent is not running, start it and it will automatically download the new plugin. If it was already running, you should run the following command in it's console to download/update the plugin:

```
> plugins update
```

Copying the JAR File

The plugin can be deployed simply by copying it to the right folder, by following these steps:

1. Stop the Jopr server and agent if they are running.

2. Copy the mobicents-slee-jopr-as-5-plugin-<*version*>.jar to the jopr-server/jbossas/server/default/deploy/rhq.ear/rhq-downloads/rhq-plugins/ directory, and it will be automatically deployed.

3. Start the Jopr server.

4. Start the Jopr agent.

After the plugin is installed, the server must be added to the managed resources. Choose **Overview** → **Auto Discovery Queue**:

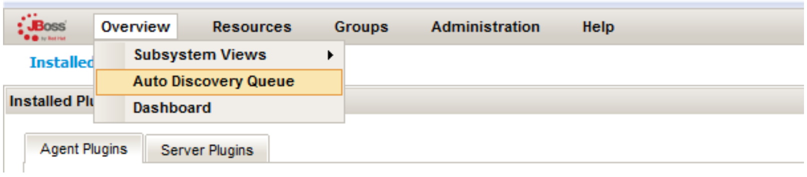


Figure 2.4. Accessing Auto Discovery Queue Menu

Note

Please make sure that the JAIN SLEE Server is running at this time, otherwise it won't be shown under Auto Discovery.

JAIN SLEE Server not showing in Auto Discovery?

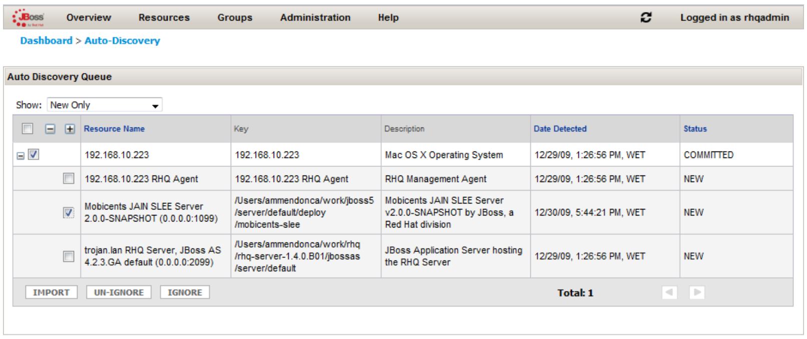
If the JAIN SLEE Server is running but not shown in the list, issue the following commands in th Jopr Agent console:

```
> plugins update
...

> discovery
```

Back in the web console, refresh the Auto Discovery view and it should be present!

Check the option referring to JAIN SLEE Server (as in the figure below), and click **IMPORT** button.



□ ⊟ ⊞	Resource Name	Key	Description	Date Detected	Status
⊟ ☑	192.168.10.223	192.168.10.223	Mac OS X Operating System	12/29/09, 1:26:56 PM, WET	COMMITTED
□	192.168.10.223 RHQ Agent	192.168.10.223 RHQ Agent	RHQ Management Agent	12/29/09, 1:26:56 PM, WET	NEW
☑	Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)	/Users/ammendonca/work/jboss5 /server/default/deploy /mobicents-slee	Mobicents JAIN SLEE Server v2.0.0-SNAPSHOT by JBoss, a Red Hat division	12/30/09, 5:44:21 PM, WET	NEW
□	trojan.lan RHQ Server, JBoss AS 4.2.3.GA default (0.0.0.0:2099)	/Users/ammendonca/work/rhq /rhq-server-1.4.0.B01/jbossas /server/default	JBoss Application Server hosting the RHQ Server	12/29/09, 1:26:56 PM, WET	NEW

Figure 2.5. Selecting the server to be imported into Jopr.

Now the server is being can be managed and monitored with Jopr.

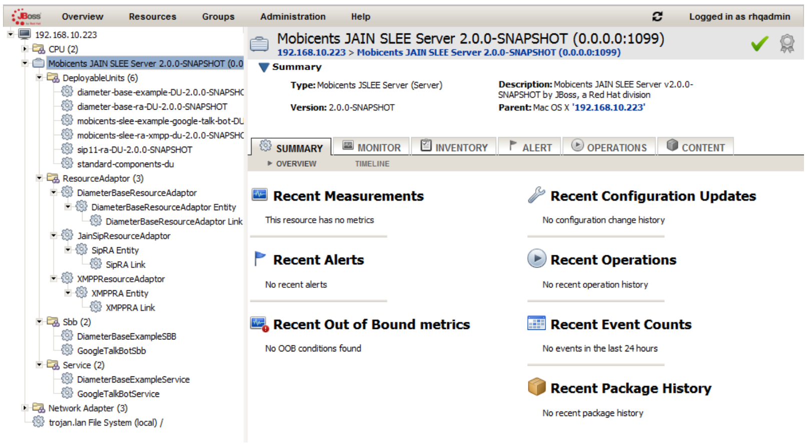
Managing Mobicents JAIN SLEE with Jopr

3.1. Jopr JAIN SLEE Plugin Main View

This section describes the main view of Jopr JAIN SLEE Plugin.

In order to access this, go to the top menu and select **Resources** → **Servers**. A list should be presented, where JAIN SLEE server is present. Click it to access server details.

You should have something similar to the following figure:



Overview
Resources
Groups
Administration
Help
Logged in as rhqadmin
192.168.10.223
CPU (2)
Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0
DeployableUnits (6)
diameter-base-example-DU-2.0.0-SNAPSHC
diameter-base-ra-DU-2.0.0-SNAPSHOT
mobicents-slee-example-google-talk-bot-DU
mobicents-slee-ra-xmpp-du-2.0.0-SNAPSHC
sip11-ra-DU-2.0.0-SNAPSHOT
standard-components-du
ResourceAdaptor (3)
DiameterBaseResourceAdaptor
DiameterBaseResourceAdaptor Entity
DiameterBaseResourceAdaptor Link
JainSipResourceAdaptor
SipRA Entity
SipRA Link
XMPPResourceAdaptor
XMPPRA Entity
XMPPRA Link
Sbb (2)
DiameterBaseExampleSBB
GoogleTalkBotSbb
Service (2)
DiameterBaseExampleService
GoogleTalkBotService
Network Adapter (3)
trojan.lan File System (local) /
Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)
192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)
Summary
SUMMARY
MONITOR
INVENTORY
ALERT
OPERATIONS
CONTENT
OVERVIEW
TIMELINE
Recent Measurements
Recent Configuration Updates
Recent Alerts
Recent Operations
Recent Out of Bound metrics
Recent Event Counts
Recent Package History

Type: Mobicents JSLEE Server (Server)

Version: 2.0.0-SNAPSHOT

Description: Mobicents JAIN SLEE Server v2.0.0-SNAPSHOT by JBoss, a Red Hat division

Parent: Mac OS X '192.168.10.223'

This resource has no metrics

No configuration change history

No recent alerts

No recent operation history

No OOB conditions found

No events in the last 24 hours

No recent package history

Figure 3.1. Jopr JAIN SLEE Plugin Main View.

As you can see in the left side menu you have the following components that can be managed by Jopr:

- Deployable Units
- Resource Adaptors
- SBBs
- Services

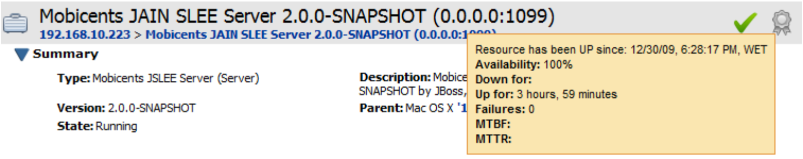
Each one will be described and explained in it's own section.

Not a component of JAIN SLEE itself, but as a crucial part of it's core is the Event Router and it's Executors. This component is also described in it's own section.

3.2. Managing and Monitoring JAIN SLEE Container

3.2.1. View JAIN SLEE Server Version and State

In the top, at the summary section the Type, Version and State of the server is shown.



Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)

192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1

Summary

Type: Mobicents JSLEE Server (Server)

Version: 2.0.0-SNAPSHOT

State: Running

Description: Mobice
SNAPSHOT by JBoss,

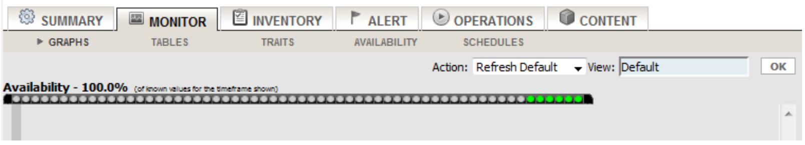
Parent: Mac OS X '1

Resource has been UP since: 12/30/09, 6:28:17 PM, WET
Availability: 100%
Down for:
Up for: 3 hours, 59 minutes
Failures: 0
MTBF:
MTTR:

Figure 3.2. Jopr JAIN SLEE Summary View.

3.2.2. Monitor JAIN SLEE Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the server over time.



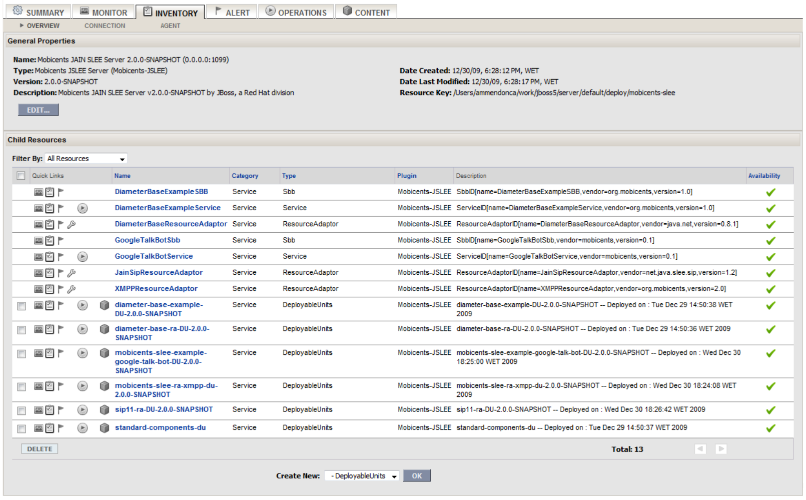
Action: Refresh Default View: Default

Availability - 100.0% (of known values for the timeframe shown)

Figure 3.3. Jopr JAIN SLEE Plugin Availability View.

3.2.3. View installed JAIN SLEE Components

By accessing the **INVENTORY** tab a table presenting the installed components and deployable units is shown.



General Properties

Name: Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)
Type: Mobicents JSLEE Server (Mobicents-JSLEE)
Version: 2.0.0-SNAPSHOT
Description: Mobicents JAIN SLEE Server v2.0.0-SNAPSHOT by JBoss, a Red Hat division

Date Created: 12/30/09, 6:28:12 PM, WET
Date Last Modified: 12/30/09, 6:28:17 PM, WET
Resource Key: /Users/ammendonca/work/jboss5/server/default/deploy/mobicents-slee

Child Resources

Filter By: All Resources

Quick Links	Name	Category	Type	Plugin	Description	Availability
	DiameterBaseExampleSBB	Service	Sbb	Mobicents-JSLEE	SbbID[name=DiameterBaseExampleSBB,vendor=org.mobicents,version=1.0]	✓
	DiameterBaseExampleService	Service	Service	Mobicents-JSLEE	ServiceID[name=DiameterBaseExampleService,vendor=org.mobicents,version=1.0]	✓
	DiameterBaseResourceAdaptor	Service	ResourceAdaptor	Mobicents-JSLEE	ResourceAdaptorID[name=DiameterBaseResourceAdaptor,vendor=java.net,version=0.8.1]	✓
	GoogleTalkBotSbb	Service	Sbb	Mobicents-JSLEE	SbbID[name=GoogleTalkBotSbb,vendor=mobicents,version=0.1]	✓
	GoogleTalkBotService	Service	Service	Mobicents-JSLEE	ServiceID[name=GoogleTalkBotService,vendor=mobicents,version=0.1]	✓
	JainSipResourceAdaptor	Service	ResourceAdaptor	Mobicents-JSLEE	ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]	✓
	XMPPResourceAdaptor	Service	ResourceAdaptor	Mobicents-JSLEE	ResourceAdaptorID[name=XMPPResourceAdaptor,vendor=org.mobicents,version=2.0]	✓
□	diameter-base-example-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	diameter-base-example-DU-2.0.0-SNAPSHOT -- Deployed on : Tue Dec 29 14:50:38 WET 2009	✓
□	diameter-base-ra-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	diameter-base-ra-DU-2.0.0-SNAPSHOT -- Deployed on : Tue Dec 29 14:50:36 WET 2009	✓
□	mobicents-slee-example-google-talk-bot-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	mobicents-slee-example-google-talk-bot-DU-2.0.0-SNAPSHOT -- Deployed on : Wed Dec 30 18:25:00 WET 2009	✓
□	mobicents-slee-ra-xmpp-du-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	mobicents-slee-ra-xmpp-du-2.0.0-SNAPSHOT -- Deployed on : Wed Dec 30 18:24:08 WET 2009	✓
□	sip11-ra-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	sip11-ra-DU-2.0.0-SNAPSHOT -- Deployed on : Wed Dec 30 18:26:42 WET 2009	✓
□	standard-components-du	Service	DeployableUnits	Mobicents-JSLEE	standard-components-du -- Deployed on : Tue Dec 29 14:50:37 WET 2009	✓

Total: 13

Create New: - DeployableUnits

Figure 3.4. Jopr JAIN SLEE Plugin Inventory View.

Components deployed outside Jopr

If components are installed outside Jopr, after Jopr Agent is started, the list of installed components will not be updated in a *reasonable* time.

For performance reasons, Jopr servers childs (such as Deployable Units, Resource Adaptors, SBBs and Services in SLEE) are checked for changes every 24 hours.

To force a service discovery, there are two options:

- In the Jopr web app, right-click the root item (Platform) of the JAIN SLEE Server and select **Execute Operation** → **Manual Autodiscovery**. Select **Yes** in **Detailed Discovery** and run the discovery by clicking the **SCHEDULE** button.
- In the Jopr agent console, run a full discovery, by issuing the following command:

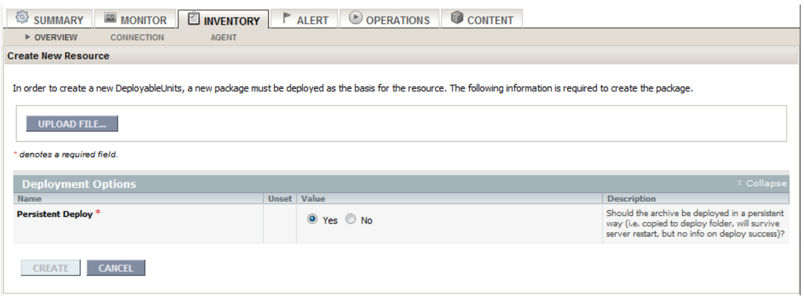
```
> discovery -f
```

3.2.4. Install/Uninstall JAIN SLEE Deployable Units

In the **INVENTORY** tab it's possible to Install/Uninstall JAIN SLEE Deployable Units.

3.2.4.1. Install JAIN SLEE Deployable Units

In order to install a deployable unit, select **DeployableUnits** on the **Create New:** option box and click **OK**, the following page will be shown:



Create New Resource

In order to create a new DeployableUnits, a new package must be deployed as the basis for the resource. The following information is required to create the package.

* denotes a required field.

Deployment Options

Name	Unset	Value	Description
Persistent Deploy *		● Yes ○ No	Should the archive be deployed in a persistent way (i.e. copied to deploy folder, will survive server restart, but no info on deploy success)?

Figure 3.5. Installing a Deployable Unit with Jopr JAIN SLEE Plugin.

Click the **UPLOAD FILE...** button, a new browser window will be shown to upload the JAIN SLEE Deployable Unit. Click **Add...**, select the Deployable Unit to be installed, click **Upload** to confirm and close this browser window.

Warning

You browser might ask you if you want to resend the information. Please accept it. Google Chrome browser may not support this operation!

Back on the main window for installing the DU, select either to do a persistent deploy (copy file to deploy folder) or not (simply call `install` on the DU).

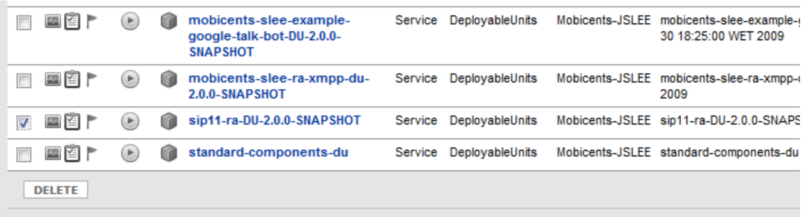
Persistent vs JMX Deployment

When doing a persistent deployment, further installation actions are automatically performed by the Mobicents JAIN SLEE Deployer. If the DU is not deployed in a persistent way (JMX), component installation actions (such as Resource Adaptor entities creation, activation, Service activation) should be performed manually.

To confirm the deployment, click **CREATE**. The success of the operation should be displayed on the main view.

3.2.4.2. Uninstall JAIN SLEE Deployable Units

To uninstall a deployable unit, simply check the desired installed deployable unit on the **INVENTORY** tab and press **DELETE**. A confirmation dialog be displayed. The success of the operation will be shown on the main view.



□	mobicents-slee-example-google-talk-bot-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	mobicents-slee-example- 30 18:25:00 WET 2009
□	mobicents-slee-ra-xmpp-du-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	mobicents-slee-ra-xmpp- 2009
☑	sip11-ra-DU-2.0.0-SNAPSHOT	Service	DeployableUnits	Mobicents-JSLEE	sip11-ra-DU-2.0.0-SNAPS
□	standard-components-du	Service	DeployableUnits	Mobicents-JSLEE	standard-components-du

Figure 3.6. Uninstalling a Deployable Unit with Jopr JAIN SLEE Plugin.

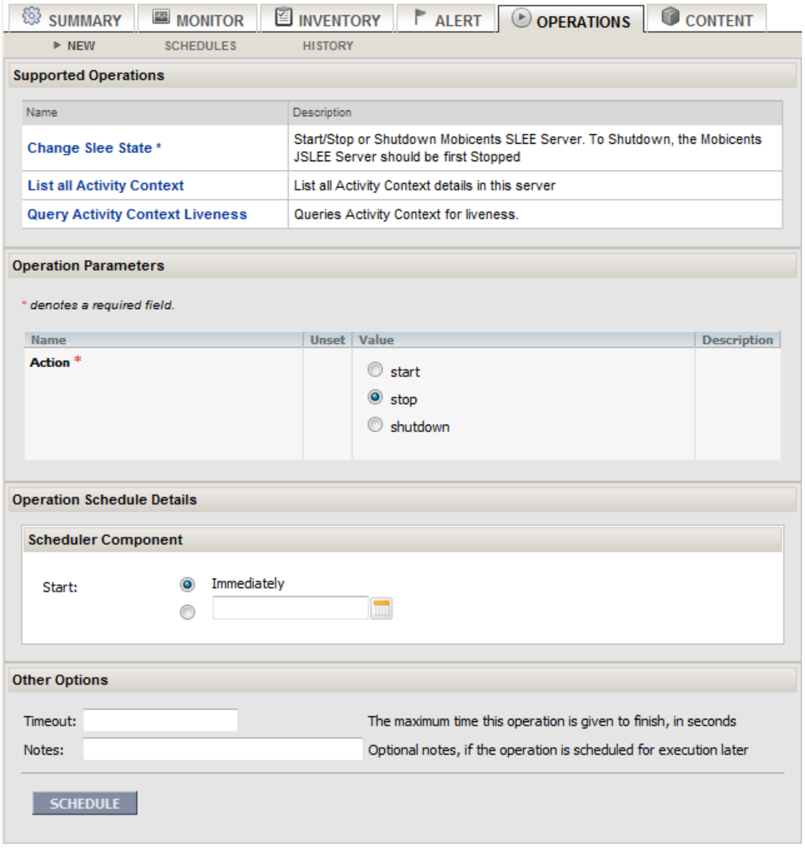
Persistent vs JMX Undeployment

When undeploying a DU it's automatically detected either if it is to do a Persistent (file is deleted from the deploy folder!) or a non Persistent undeployment, where only the `uninstall` command is called on the DU, requiring prior uninstalling actions (eg: remove RA Links and Entities) to have already been performed.

3.2.5. Start/Stop/Shutdown Mobicents JAIN SLEE Server

Under the **OPERATIONS** tab, an operation for Changing the JAIN SLEE container state can be found: **Change Slee State**.

Clicking on the operation, a new pane opens so the parameters can be selected:



Supported Operations

Name	Description
Change Slee State *	Start/Stop or Shutdown Mobicents SLEE Server. To Shutdown, the Mobicents JSLEE Server should be first Stopped
List all Activity Context	List all Activity Context details in this server
Query Activity Context Liveness	Queries Activity Context for liveness.

Operation Parameters

* denotes a required field.

Name	Unset	Value	Description
Action *		○ start ● stop ○ shutdown	

Operation Schedule Details

Scheduler Component

Start: ● Immediately
○

Other Options

Timeout: The maximum time this operation is given to finish, in seconds

Notes: Optional notes, if the operation is scheduled for execution later

Figure 3.7. Changing JAIN SLEE State.

Change Slee State Parameters

Action (required)

- start - Changes SLEE State to RUNNING state.
- stop - Changes SLEE State to STOPPED state.
- shutdown - Completely shutsdown SLEE. Requires to be in STOPPED state.

Note

As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

3.2.6. List All Activity Contexts

For monitoring purposes it is possible to have all JAIN SLEE Activity Contexts listed. This can be done with **List all Activity Context** operation, in **OPERATIONS** tab.

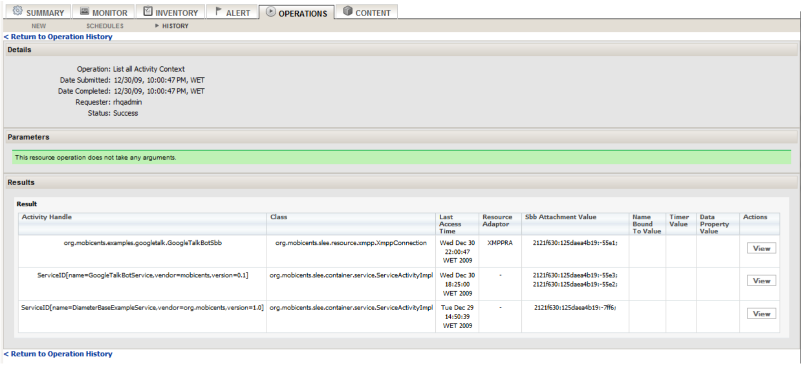
This action takes no parameters.

Note

As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

The result is presented in the following form:



Details

Operation: List all Activity Context
Date Submitted: 12/30/09, 10:00:47 PM, WET
Date Completed: 12/30/09, 10:00:47 PM, WET
Requester: rhqadmin
Status: Success

Parameters

This resource operation does not take any arguments.

Results

Result

Activity Handle	Class	Last Access Time	Resource Adaptor	Sbb Attachment Value	Name Bound To Value	Timer Value	Data Property Value	Actions
org.mobicents.examples.googletalk.GoogleTalkBotSbb	org.mobicents.slee.resource.xmpp.XmppConnection	Wed Dec 30 22:00:47 WET 2009	XMPPRA	2121f630:125daea4b19:-55e1;				View

ServiceID[name=GoogleTalkBotService,vendor=mobicents,version=0.1]	org.mobicents.slee.container.service.ServiceActivityImpl	Wed Dec 30 18:25:00 WET 2009	-	2121f630:125daea4b19:-55e3;
				2121f630:125daea4b19:-55e2;
								View

ServiceID[name=DiameterBaseExampleService,vendor=org.mobicents,version=1.0]	org.mobicents.slee.container.service.ServiceActivityImpl	Tue Dec 29 14:50:39 WET 2009	-	2121f630:125daea4b19:-7ff6;				View

Figure 3.8. Sample result of invoking List All Activity Contexts.

3.2.7. Query Activity Context Liveness

In order to clean dead Activities, it's possible to query Activity Context for their liveness. This can be done using **Query Activity Context Liveness** operation, in **OPERATIONS** tab.

This action takes no parameters.

Note

As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

3.2.8. Change Container Logging Configuration

It is possible to switch the JAIN SLEE container logging configuration for better fitting the needs of the container. This can be done using **Change Global Logging Profile** operation, in **OPERATIONS** tab. It can be switched between three levels:

Logging Configuration Levels

Level

- DEFAULT: Regular logging, at INFO level, displaying most user-related messages;
- DEBUG: More verbose logging, mostly using DEBUG/TRACE level, displaying message of interest for developers;
- PRODUCTION: Low verbosity and async logging, mostly in WARN level, for systems in production so that logging does impact performance.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

> **Note**
>
> This changes the whole container logging configuration, so it will affect also logging for other running applications in the container, besides JAIN SLEE.

3.2.9. Edit Container Logging Configurations

It is possible to customize the above mentioned logging configurations, or also the currently in use container logging configuration. To perform this action, two operations should be executed:

First, to retrieve the current logging configuration, the **Get Logging Profile Configuration** operation, in **OPERATIONS** tab should be used. One of the following options should be selected:

Logging Configuration Levels

Level

- CURRENT: will retrieve the logging configuration currently in use by the container (can be a custom one);

- DEFAULT: will retrieve the DEFAULT logging configuration;
- DEBUG: will retrieve the DEBUG logging configuration;
- PRODUCTION: will retrieve the PRODUCTION logging configuration;

After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result, which is the actual logging configuration.

The result of the operation should be copied into your favourite XML editor and edited as desired.

Logging configuration size limitation

Currently there's a limitation on 2000 characters for the logging files, so caution should be used not to pass this limit when editing.

When the editing is completed, the **Update Logging Profile Configuration** operation should be used to update the logging configuration contents.

Select the desired configuration to be updated from the similar list as presented above, and paste the edited contents in the **Contents** field of the operation.

Note

As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

Note

Editing the DEFAULT, DEBUG or PRODUCTION configurations does not apply them, even if they are currently being used. It only edits the configuration template. To edit the configuration being used, select the CURRENT configuration.

3.2.10. View Event Router Statistics

The Jopr JAIN SLEE platform enables you to gather statistics on it's Event Router. With the **View Event Router Statistics** operation, in **OPERATIONS** tab it's possible to view them in a table form.

The statistics can be filtered by the following levels:

Event Router Statistics Filters

Filter

- Global: Shows the container global statistics for all executors and event types (eg: container total number of events routed, activities mapped, etc.);
- Executors: Shows the statistics per each executor (eg: activities mapped per executor, average routing time per executor, idle time per executor, etc.);
- Event Types: Shows the statistics per each event type deployed (eg: average routing time per event type, number of events of that type routed, etc.);
- Executors and Event Types: Shows the combined statistics of Executors with Event Types (eg: number of events of certain type router per each executor, average time for routing a certain event type by each executor, total time spent per each executor routing certain event types, etc.);
- All: All the above together.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

3.2.11. Managing Congestion Control

The Jopr JAIN SLEE platform provides congestion control mechanisms to prevent the server to become unresponsive under very high load.

The following parameters are available for configuration and the correspondent set operations (CC-Set<parameter>) can be found under **OPERATIONS** tab:

Congestion Control Parameters

Parameter

- *MinFreeMemoryToTurnOn*: The minimum memory (percentage) that should be available to turn on congestion control;
- *MinFreeMemoryToTurnOff*: The minimum memory (percentage) that should be available to turn off congestion control;

- *PeriodBetweenChecks*: The period in seconds to check if congestion control state should change. 0 means congestion control is off;
- *RefuseStartActivity*: Defines if the start of activity should be refused, when congestion control is on;
- *RefuseFireEvent*: Defines if the firing of an event should be refused, when congestion control is on;

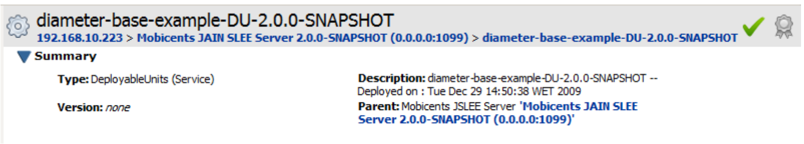
> **Note**
>
> As any operation on Jopr, these operations can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on its name shows the result(s).

3.3. Managing and Monitoring JAIN SLEE Deployable Units

3.3.1. View JAIN SLEE Deployable Unit Summary

In the top, at the summary section, some details regarding the Deployable Unit can be seen, such as the Type, Parent Component (JSLEE Server) and the date it was deployed on, in **Description**.



diameter-base-example-DU-2.0.0-SNAPSHOT
192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > diameter-base-example-DU-2.0.0-SNAPSHOT

Summary

Type: DeployableUnits (Service)

Version: *none*

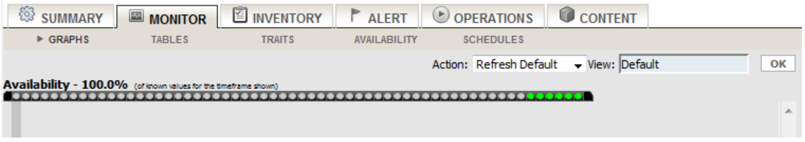
Description: diameter-base-example-DU-2.0.0-SNAPSHOT -- Deployed on : Tue Dec 29 14:50:38 WET 2009

Parent: Mobicents JSLEE Server 'Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)'

Figure 3.9. Jopr JAIN SLEE Deployable Unit Summary View.

3.3.2. Monitor JAIN SLEE DU Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the deployable unit over time.



Action: Refresh Default View: Default

Availability - 100.0% (of known values for the timeframe shown)

Figure 3.10. Jopr JAIN SLEE Deployable Unit Availability View.

Deployable Unit Availability Feature

For a deployable unit, availability refers to whether the DU was or not installed. It only works for deployable units with the same name.

3.3.3. List Component IDs

The only operation available for Deployment Units is to view the IDs of the components that it contains, it can be done by executing the **List Component IDs** operation.

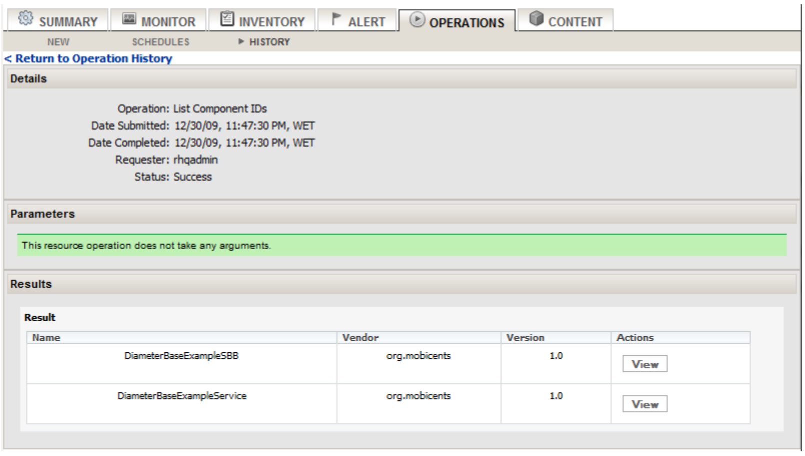
This action takes no parameters.

Note

As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

The result is presented in the following form:



Details

Operation: List Component IDs
Date Submitted: 12/30/09, 11:47:30 PM, WET
Date Completed: 12/30/09, 11:47:30 PM, WET
Requester: rhqadmin
Status: Success

Parameters

This resource operation does not take any arguments.

Results

Result

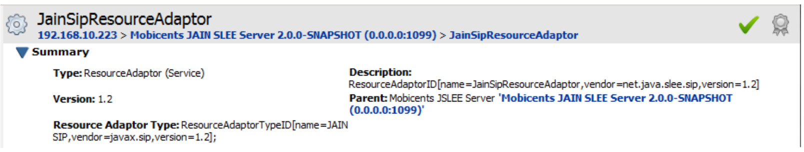
Name	Vendor	Version	Actions
DiameterBaseExampleSBB	org.mobicents	1.0	View
DiameterBaseExampleService	org.mobicents	1.0	View

Figure 3.11. Sample result of invoking List Component IDs.

3.4. Managing and Monitoring JAIN SLEE Resource Adaptors

3.4.1. View JAIN SLEE Resource Adaptor Summary

In the top, at the summary section, some details regarding the Resource Adaptor can be seen, such as the Component ID (in **Description**), the Resource Adaptor Type it implements and its Parent Component (JSLEE Server).



JainSipResourceAdaptor
192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > JainSipResourceAdaptor

Summary

Type: ResourceAdaptor (Service)

Version: 1.2

Resource Adaptor Type: ResourceAdaptorTypeID[name=JAIN SIP,vendor=javax.sip,version=1.2];

Description: ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]

Parent: Mobicents JSLEE Server 'Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)'

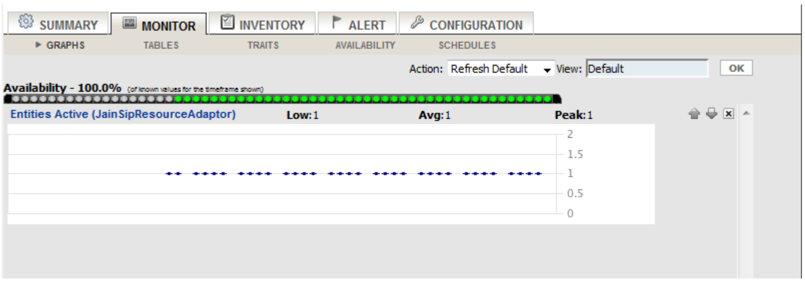
Figure 3.12. Jopr JAIN SLEE Resource Adaptor Summary View.

3.4.2. Monitor JAIN SLEE RA Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the Resource Adaptor over time.

3.4.3. Monitor JAIN SLEE RA Active Entities over time

Under the **MONITOR** tab it's also possible to view how many Entities from this RA have been active over time.



Action: Refresh Default View: Default

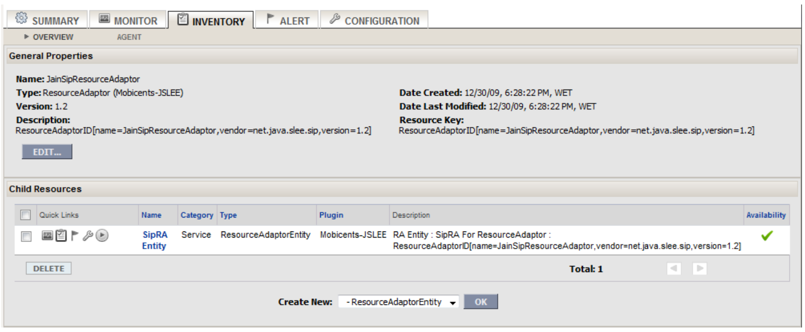
Availability - 100.0% (of known values for the timeframe shown)

Entities Active (JainSipResourceAdaptor) Low: 1 Avg: 1 Peak: 1

Figure 3.13. Monitoring JAIN SLEE Resource Adaptor Availability and Active Entities.

3.4.4. View, Create and Remove Resource Adaptor Entities

In the **INVENTORY** tab it's possible to View, Create and Remove Resource Adaptor Entities.



General Properties

Name: JainSipResourceAdaptor
Type: ResourceAdaptor (Mobicents-JSLEE)
Version: 1.2
Description:
ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]

Date Created: 12/30/09, 6:28:22 PM, WET
Date Last Modified: 12/30/09, 6:28:22 PM, WET
Resource Key:
ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]

Child Resources

□	Quick Links	Name	Category	Type	Plugin	Description	Availability
□		SipRA Entity	Service	ResourceAdaptorEntity	Mobicents-JSLEE	RA Entity : SipRA For ResourceAdaptor : ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]	✓

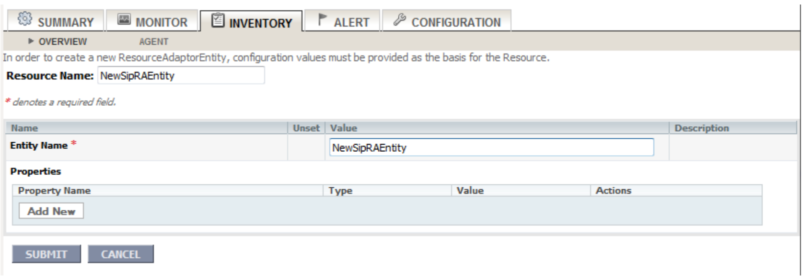
Total: 1

Create New: - ResourceAdaptorEntity

Figure 3.14. Resource Adaptor Entities Listing

3.4.4.1. Create a Resource Adaptor Entity

To create a RA Entity, select **ResourceAdaptorEntity** on the **Create New:** option box and click **OK**, the following page will be shown:



In order to create a new ResourceAdaptorEntity, configuration values must be provided as the basis for the Resource.

Resource Name: NewSipRAEntity

** denotes a required field.*

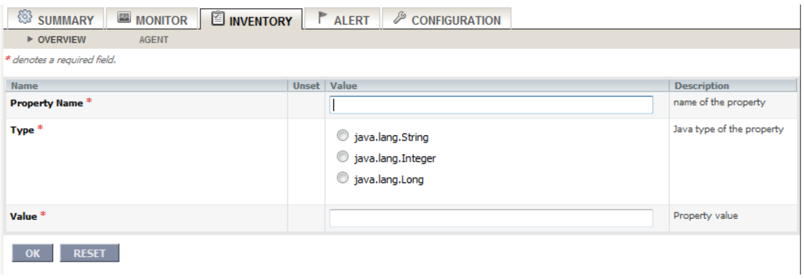
Name	Unset	Value	Description
Entity Name *		NewSipRAEntity	

Properties

Property Name	Type	Value	Actions

Figure 3.15. Create New Resource Adaptor Entity

Fill the **Resource Name** and **Entity Name** with the desired new entity name. If desired, provide extra properties for the entity in the Properties table by clicking **Add New**, which will present the following screen:



** denotes a required field.*

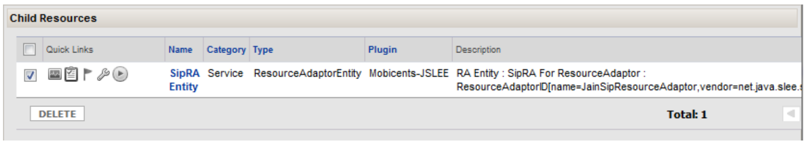
Name	Unset	Value	Description
Property Name *			name of the property
Type *		○ java.lang.String ○ java.lang.Integer ○ java.lang.Long	Java type of the property
Value *			Property value

Figure 3.16. Add Property to New Resource Adaptor Entity

After adding the properties, click the **SUBMIT** button and the new RA Entity will be created. The success of the operation will be shown in the main view.

3.4.4.2. Remove a Resource Adaptor Entity

To remove a RA Entity, check it on the **INVENTORY** tab and click the on the **DELETE** button. A confirmation dialog be displayed. The success of the operation will be shown in the main view.



Child Resources

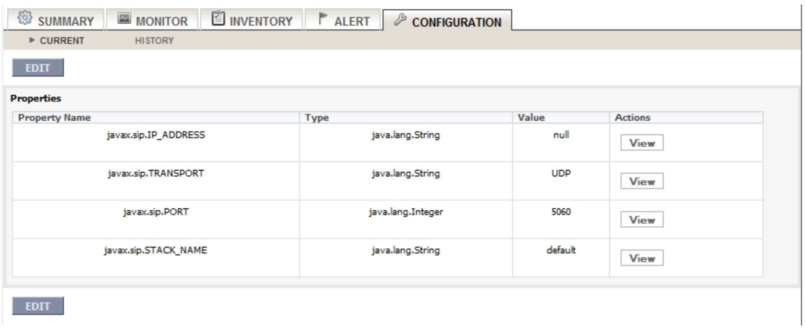
☐	Quick Links	Name	Category	Type	Plugin	Description
☑		SipRA Entity	Service	ResourceAdaptorEntity	Mobicents-JSLEE	RA Entity : SipRA For ResourceAdaptor : ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.s

Total: 1

Figure 3.17. Removing a Resource Adaptor Entity with Jopr JAIN SLEE Plugin.

3.4.5. Configure a Resource Adaptor

Under the **CONFIGURATION** tab Resource Adaptor properties can be defined, changed and or removed.



Properties

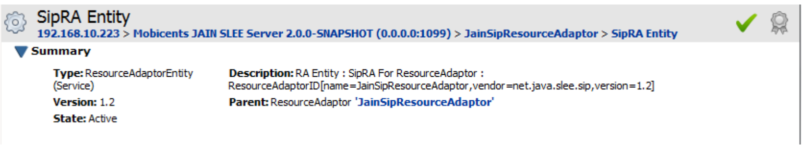
Property Name	Type	Value	Actions
javax.sip.IP_ADDRESS	java.lang.String	null	View
javax.sip.TRANSPORT	java.lang.String	UDP	View
javax.sip.PORT	java.lang.Integer	5060	View
javax.sip.STACK_NAME	java.lang.String	default	View

Figure 3.18. Configuring Resource Adaptor Properties.

3.5. Managing and Monitoring JAIN SLEE Resource Adaptor Entities

3.5.1. View Resource Adaptor Entity Summary

In the top, at the summary section, some details regarding the Resource Adaptor Entity can be seen, such as the Resource Adaptor it belongs to (in **Description**), the current State and its Parent Component (Resource Adaptor).



SipRA Entity

192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > JainSipResourceAdaptor > SipRA Entity

Summary

Type: ResourceAdaptorEntity (Service)

Version: 1.2

State: Active

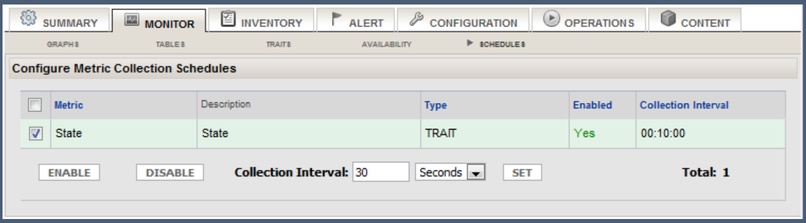
Description: RA Entity : SipRA For ResourceAdaptor : ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]

Parent: ResourceAdaptor 'JainSipResourceAdaptor'

Figure 3.19. Resource Adaptor Entity Summary View.

Change State/Metrics update interval

By default, State property (as other metric values) is updated every 10 minutes. To set it to a shorter value (minimum: 30 seconds) go to **SCHEDULES** sub-menu under **MONITOR** tab, check the desired option(s), enter the desired amount of time in the **Collection Interval** textbox, select the units (seconds, minutes, hours) and click **SET**.



Configure Metric Collection Schedules

□	Metric	Description	Type	Enabled	Collection Interval
☑	State	State	TRAIT	Yes	00:10:00

ENABLE DISABLE Collection Interval: 30 Seconds SET Total: 1

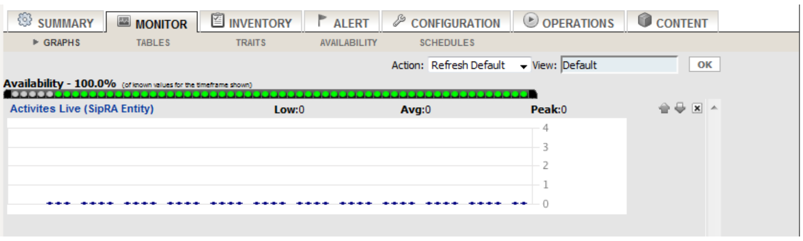
Figure 3.20. Change Collection Interval value for measurements.

3.5.2. Monitor JAIN SLEE RA Entity Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the Resource Adaptor Entity over time.

3.5.3. Monitor JAIN SLEE RA Entity Activities over time

Under the **MONITOR** tab it's also possible to view how many Activities this RA Entity is responsible for, over time.



Action: Refresh Default View: Default

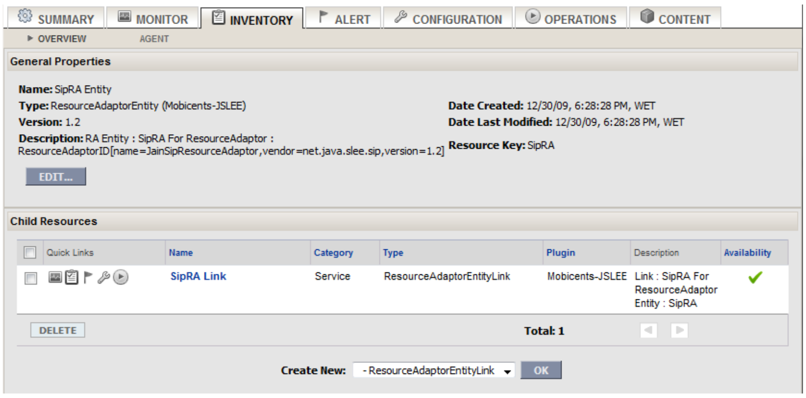
Availability - 100.0% (of known values for the timeframe shown)

Activites Live (SipRA Entity) Low:0 Avg:0 Peak:0

Figure 3.21. Monitoring JAIN SLEE Resource Adaptor Entity Availability and Activities Count.

3.5.4. View, Create and Remove Resource Adaptor Entity Links

In the **INVENTORY** tab it's possible to View, Create and Remove Resource Adaptor Entity Links.



General Properties

Name: SipRA Entity
Type: ResourceAdaptorEntity (Mobicents-JSLEE)
Version: 1.2
Description: RA Entity : SipRA For ResourceAdaptor : ResourceAdaptorID[name=JainSipResourceAdaptor,vendor=net.java.slee.sip,version=1.2]

Date Created: 12/30/09, 6:28:28 PM, WET
Date Last Modified: 12/30/09, 6:28:28 PM, WET
Resource Key: SipRA

Child Resources

□	Quick Links	Name	Category	Type	Plugin	Description	Availability
□		SipRA Link	Service	ResourceAdaptorEntityLink	Mobicents-JSLEE	Link : SipRA For ResourceAdaptor Entity : SipRA	✓

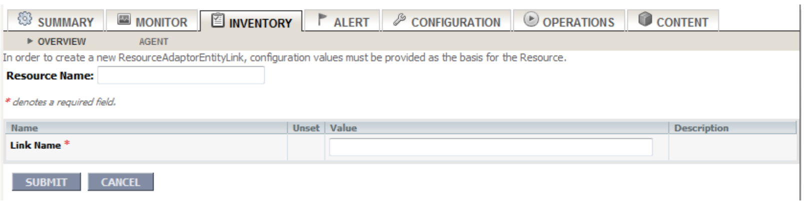
Total: 1

Create New: - ResourceAdaptorEntityLink

Figure 3.22. Resource Adaptor Entity Links Listing

3.5.4.1. Create a Resource Adaptor Entity Link

To create a RA Entity Link, select **ResourceAdaptorEntityLink** on the **Create New:** option box and click **OK**, the following page will be shown:



In order to create a new ResourceAdaptorEntityLink, configuration values must be provided as the basis for the Resource.

Resource Name:

** denotes a required field.*

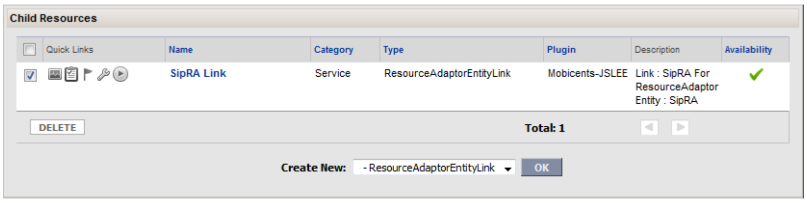
Name	Unset	Value	Description
Link Name *			

Figure 3.23. Create New Resource Adaptor Entity Link

Fill the **Resource Name** and **Link Name** with the desired new entity link name. Click the **SUBMIT** button and the new RA Entity will be created. The success of the operation will be shown in the main view.

3.5.4.2. Remove a Resource Adaptor Entity Link

To remove a RA Entity, check it on the **INVENTORY** tab and click the on the **DELETE** button. A confirmation dialog be displayed. The success of the operation will be shown in the main view.



Child Resources

□	Quick Links	Name	Category	Type	Plugin	Description	Availability
☑		SipRA Link	Service	ResourceAdaptorEntityLink	Mobicents-JSLEE	Link : SipRA For ResourceAdaptor Entity : SipRA	✓

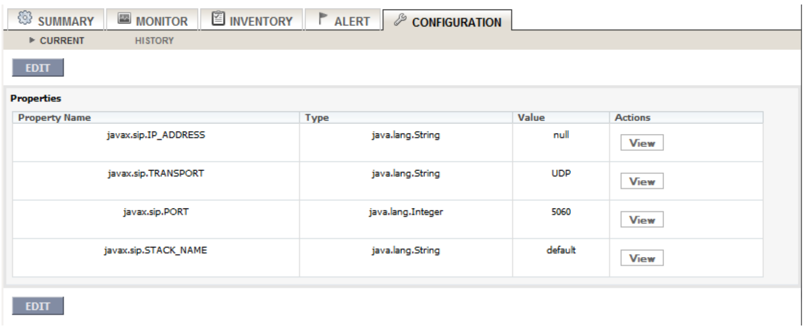
Total: 1

Create New: - ResourceAdaptorEntityLink

Figure 3.24. Removing a Resource Adaptor Entity Link with Jopr JAIN SLEE Plugin.

3.5.5. Configure a Resource Adaptor Entity

Under the **CONFIGURATION** tab Resource Adaptor Entity properties can be defined, changed and or removed.



Properties

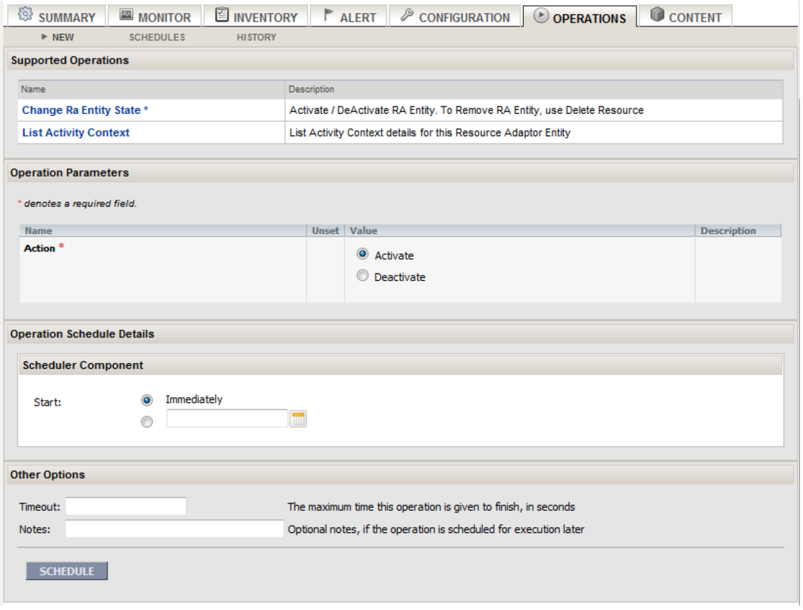
Property Name	Type	Value	Actions
javax.sip.IP_ADDRESS	java.lang.String	null	View
javax.sip.TRANSPORT	java.lang.String	UDP	View
javax.sip.PORT	java.lang.Integer	5060	View
javax.sip.STACK_NAME	java.lang.String	default	View

Figure 3.25. Configuring a Resource Adaptor Entity with Jopr JAIN SLEE Plugin.

3.5.6. Activate/Deactivate RA Entity

In the **OPERATIONS** tab, an operation to Activate/Deactivate the RA Entity is available, named **Change Ra Entity State**.

Clicking on the operation, a new pane opens so the parameters can be selected:



Supported Operations

Name	Description
Change Ra Entity State *	Activate / DeActivate RA Entity. To Remove RA Entity, use Delete Resource
List Activity Context	List Activity Context details for this Resource Adaptor Entity

Operation Parameters

* denotes a required field.

Name	Unset	Value	Description
Action *		● Activate ○ Deactivate	

Operation Schedule Details

Scheduler Component

Start: ● Immediately
○

Other Options

Timeout: The maximum time this operation is given to finish, in seconds

Notes: Optional notes, if the operation is scheduled for execution later

Figure 3.26. Changing the Resource Adaptor Entity State.

Change Ra Entity State Parameters

Action (required)

- Activate - Activate the RA Entity.

- Deactivate - Deactivate the RA Entity.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

3.5.7. List Activity Context

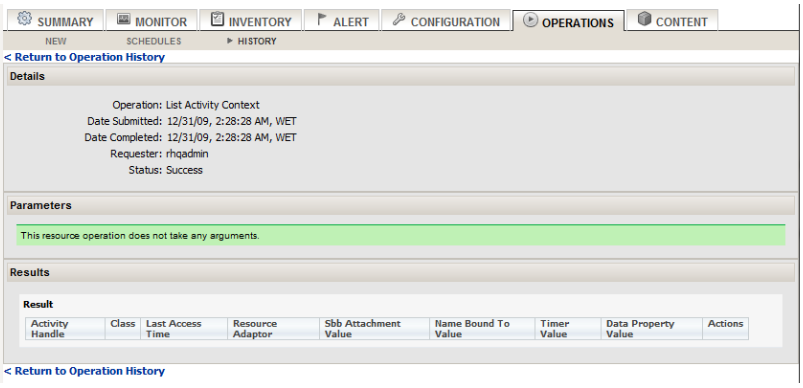
It's possible to view the Activity Contexts owned by this RA Entity, with the operation named **List Activity Context** on the **OPERATIONS** tab.

This action takes no parameters.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

The result is presented in the following form:



Details

Operation: List Activity Context
Date Submitted: 12/31/09, 2:28:28 AM, WET
Date Completed: 12/31/09, 2:28:28 AM, WET
Requester: rhqadmin
Status: Success

Parameters

This resource operation does not take any arguments.

Results

Result

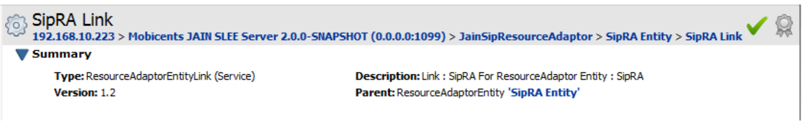
Activity Handle	Class	Last Access Time	Resource Adaptor	Sbb Attachment Value	Name Bound To Value	Timer Value	Data Property Value	Actions

Figure 3.27. Sample result of invoking List Activity Context on RA Entity.

3.6. Managing and Monitoring JAIN SLEE Resource Adaptor Entity Links

3.6.1. View Resource Adaptor Entity Link Summary

In the top, at the summary section, some details regarding the Resource Adaptor Entity Link can be seen, such as the Resource Adaptor Entity it belongs to (in **Description**).



SipRA Link

192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > JainSipResourceAdaptor > SipRA Entity > SipRA Link

Summary

Type: ResourceAdaptorEntityLink (Service)

Version: 1.2

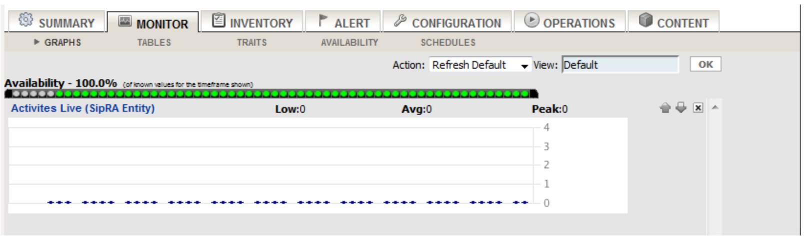
Description: Link : SipRA For ResourceAdaptor Entity : SipRA

Parent: ResourceAdaptorEntity 'SipRA Entity'

Figure 3.28. Resource Adaptor Entity Link Summary View.

3.6.2. Monitor JAIN SLEE RA Entity Link Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the Resource Adaptor Entity Link over time.



Action: Refresh Default View: Default

Availability - 100.0% (of known values for the timeframe shown)

Activites Live (SipRA Entity) Low:0 Avg:0 Peak:0

Figure 3.29. Monitoring JAIN SLEE Resource Adaptor Entity Link Availability.

3.6.3. List SBBs bound to RA Entity Link

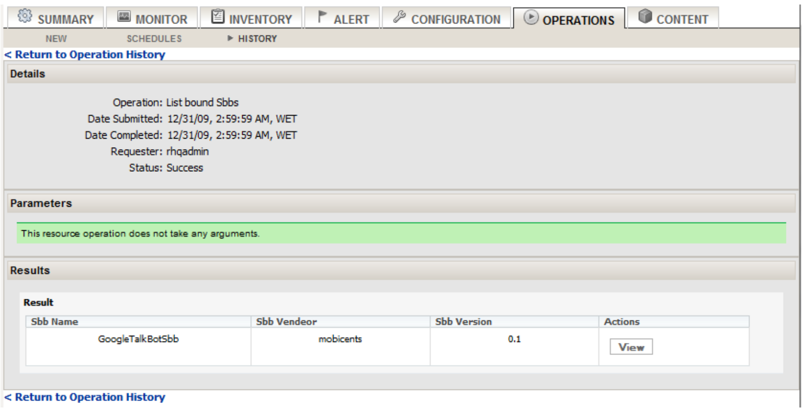
In the **OPERATIONS** tab, an operation to list all SBBs bound to a specific link is available, named **List bound Sbbs**.

This action takes no parameters.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

The result is presented in the following form:



< Return to Operation History

Details

Operation: List bound Sbbs
Date Submitted: 12/31/09, 2:59:59 AM, WET
Date Completed: 12/31/09, 2:59:59 AM, WET
Requester: rhqadmin
Status: Success

Parameters

This resource operation does not take any arguments.

Results

Result

Sbb Name	Sbb Vendeor	Sbb Version	Actions
GoogleTalkBotSbb	mobicents	0.1	View

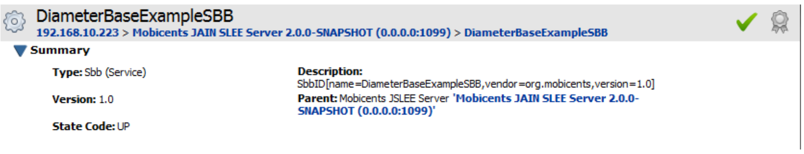
< Return to Operation History

Figure 3.30. Sample result of invoking List bound Sbbs on RA Entity Link.

3.7. Managing and Monitoring JAIN SLEE Service Building Blocks (SBBs)

3.7.1. View SBB Summary

In the top, at the summary section, some details regarding the SBB can be seen, such as the Component ID (in **Description**), the Version, the State Code (UP/DOWN) and its Parent Component (JSLEE Server).



DiameterBaseExampleSBB
192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > DiameterBaseExampleSBB

Summary

Type: Sbb (Service)

Version: 1.0

State Code: UP

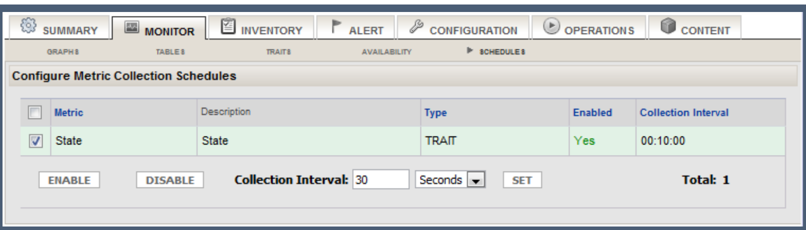
Description:
SbbID[name=DiameterBaseExampleSBB,vendor=org.mobicents,version=1.0]

Parent: Mobicents JSLEE Server 'Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)'

Figure 3.31. SBB Summary View.

Change State/Metrics update interval

By default, State property (as other metric values) is updated every 10 minutes. To set it to a shorter value (minimum: 30 seconds) go to **SCHEDULES** sub-menu under **MONITOR** tab, check the desired option(s), enter the desired amount of time in the **Collection Interval** textbox, select the units (seconds, minutes, hours) and click **SET**.



Configure Metric Collection Schedules

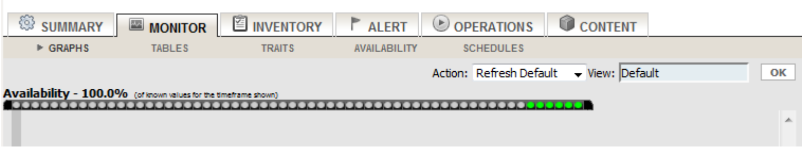
□	Metric	Description	Type	Enabled	Collection Interval
☑	State	State	TRAIT	Yes	00:10:00

ENABLE DISABLE **Collection Interval:** 30 Seconds SET **Total: 1**

Figure 3.32. Change Collection Interval value for measurements.

3.7.2. Monitor JAIN SLEE SBB Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the SBB over time.



Action: Refresh Default View: Default

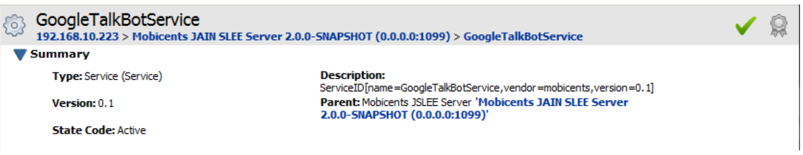
Availability - 100.0% (of known values for the timeframe shown)

Figure 3.33. Monitoring JAIN SLEE SBB Availability.

3.8. Managing and Monitoring JAIN SLEE Services

3.8.1. View Service Summary

In the top, at the summary section, some details regarding the Service can be seen, such as the Component ID (in **Description**), the Version, the State and its Parent Component (JSLEE Server).



GoogleTalkBotService

192.168.10.223 > Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099) > GoogleTalkBotService

Summary

Type: Service (Service)

Version: 0.1

State Code: Active

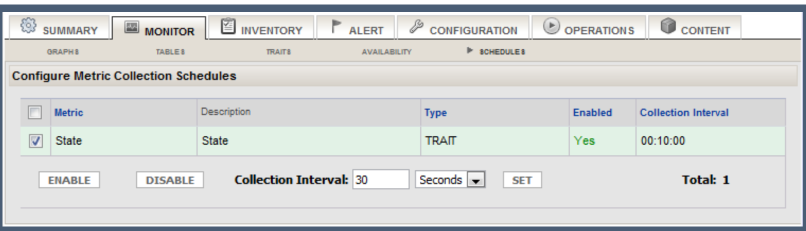
Description:
ServiceID[name=GoogleTalkBotService,vendor=mobicents,version=0.1]

Parent: Mobicents JSLEE Server 'Mobicents JAIN SLEE Server 2.0.0-SNAPSHOT (0.0.0.0:1099)'

Figure 3.34. Service Summary View.

Change State/Metrics update interval

By default, State property (as other metric values) is updated every 10 minutes. To set it to a shorter value (minimum: 30 seconds) go to **SCHEDULES** sub-menu under **MONITOR** tab, check the desired option(s), enter the desired amount of time in the **Collection Interval** textbox, select the units (seconds, minutes, hours) and click **SET**.



Configure Metric Collection Schedules

☐	Metric	Description	Type	Enabled	Collection Interval
☑	State	State	TRAIT	Yes	00:10:00

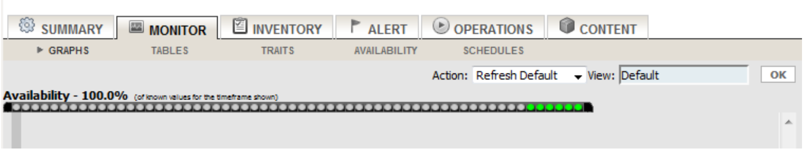
Collection Interval: 30 Seconds

Total: 1

Figure 3.35. Change Collection Interval value for measurements.

3.8.2. Monitor JAIN SLEE Service Availability over time

By accessing the **MONITOR** tab it's possible to see a graph showing the availability of the Service over time.



Action: Refresh Default View: Default

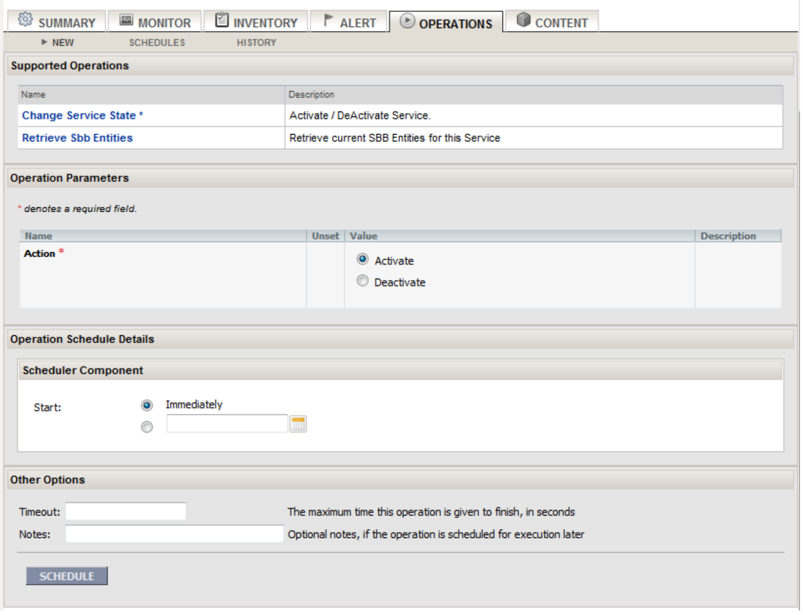
Availability - 100.0% (of known values for the timeframe shown)

Figure 3.36. Monitoring JAIN SLEE Service Availability.

3.8.3. Activate/Deactivate a JAIN SLEE Service

Under the **OPERATIONS** tab, an operation for changing the state of the service can be found: **Change Service State**.

Clicking on the operation, a new pane opens so the parameters can be selected:



Supported Operations

Name	Description
Change Service State *	Activate / DeActivate Service.
Retrieve Sbb Entities	Retrieve current SBB Entities for this Service

Operation Parameters

* denotes a required field.

Name	Unset	Value	Description
Action *		● Activate ○ Deactivate	

Operation Schedule Details

Scheduler Component

Start: ● Immediately
○

Other Options

Timeout: The maximum time this operation is given to finish, in seconds

Notes: Optional notes, if the operation is scheduled for execution later

Figure 3.37. Changing Service State.

Change Service State Parameters

Action (required)

- Activate - Activate the Service.
- Deactivate - Deactivate the Service.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

3.8.4. Retrieve SBB Entities

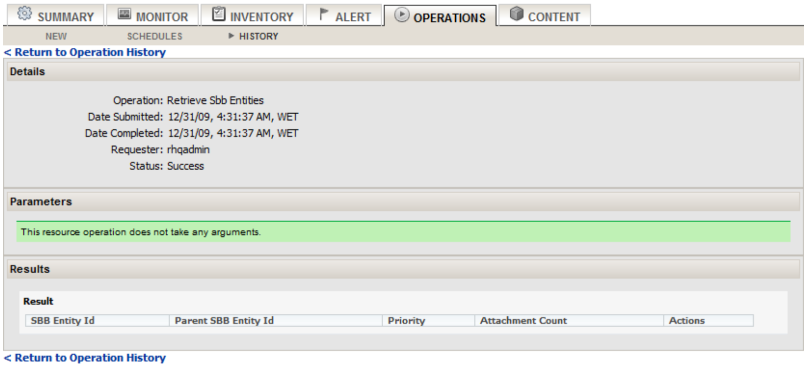
In the **OPERATIONS** tab, an operation to list all SBB Entities belonging to a specific service is available, named **Retrieve Sbb Entities**.

This action takes no parameters.

> **Note**
>
> As any operation on Jopr, this operation can be scheduled for a later time, have a defined timeout and/or include additional notes.

Clicking on the **SCHEDULE** executes (if **Immediately** is selected) or schedules the operation. After being executed, it's shown in **HISTORY**, **Completed Operations**. Clicking on it shows the result(s).

The result is presented in the following form:



Details

Operation: Retrieve Sbb Entities
Date Submitted: 12/31/09, 4:31:37 AM, WET
Date Completed: 12/31/09, 4:31:37 AM, WET
Requester: rhqadmin
Status: Success

Parameters

This resource operation does not take any arguments.

Results

Result

SBB Entity Id	Parent SBB Entity Id	Priority	Attachment Count	Actions

Figure 3.38. Sample result of invoking Retrieve Sbb Entities on Service.

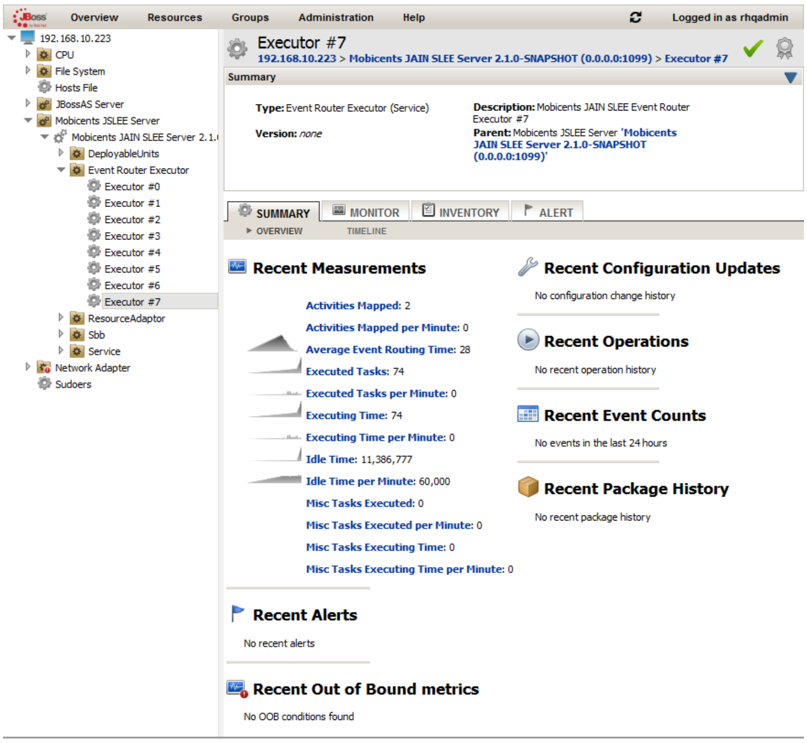
3.9. Monitoring JAIN SLEE Event Router Executors

Event Router Executors are the heart of the JAIN SLEE Core, they are responsible for executing critical tasks for the container.

In order to have a better view over the JAIN SLEE container health, a few metrics are made available in this section.

3.9.1. View Event Router Executors Summary

In the **SUMMARY** tab, some of the metrics for the executors are shown along with the corresponding graphs. Also, latest Out-of-Bound metrics and Alerts are displayed.



Executor #7

192.168.10.223 > Mobicents JAIN SLEE Server 2.1.0-SNAPSHOT (0.0.0.0:1099) > Executor #7

Summary

Type: Event Router Executor (Service)

Version: *none*

Description: Mobicents JAIN SLEE Event Router Executor #7

Parent: Mobicents JSLEE Server 'Mobicents JAIN SLEE Server 2.1.0-SNAPSHOT (0.0.0.0:1099)'

Recent Measurements

Activities Mapped: 2
Activities Mapped per Minute: 0
Average Event Routing Time: 28
Executed Tasks: 74
Executed Tasks per Minute: 0
Executing Time: 74
Executing Time per Minute: 0
Idle Time: 11,386,777
Idle Time per Minute: 60,000
Misc Tasks Executed: 0
Misc Tasks Executed per Minute: 0
Misc Tasks Executing Time: 0
Misc Tasks Executing Time per Minute: 0

Recent Configuration Updates

No configuration change history

Recent Operations

No recent operation history

Recent Event Counts

No events in the last 24 hours

Recent Package History

No recent package history

Recent Alerts

No recent alerts

Recent Out of Bound metrics

No OOB conditions found

Figure 3.39. Jopr JAIN SLEE Event Router Executors Summary View.

3.9.2. Monitor Event Router Executors metrics

By accessing the **MONITOR** tab it's possible to see a lot of metrics for each executor:

- *Activities Mapped*: Total activities mapped for all the event router executors
- *Activities Mapped per Minute*: Total activities mapped for all the event router executors
- *Average Event Routing Time*: Average time spent to route one event (in ms)
- *Executed Tasks*: Number of tasks executed, which is the sum of the events routed for each EventTypeID, and also the misc tasks executed

- *Executed Tasks per Minute*: Same as above, per minute
- *Executing Time*: Time spent on executing tasks, which is the sum of the time spent routing events for each EventTypeID, and also the time spent executing the misc tasks.
- *Executing Time per Minute*: Same as above, per minute
- *Idle Time*: Total time the executor was idle, i.e., not executing tasks
- *Idle Time per Minute*: Same as above, per minute
- *Misc Tasks Executed*: Number of misc tasks executed
- *Misc Tasks Executed per Minute*: Same as above, per minute
- *Misc Tasks Executing Time*: Time spent executing misc tasks.
- *Misc Tasks Executing Time per Minute*: Same as above, per minute

This metrics can also be seen in a table, in the **TABLES** sub-menu, in **MONITOR** tab.

* per Minute Metrics

The "... per Minute" metrics are calculated by Jopr so, they may not be accurate for container lifetime, ie, they are based on values gathered while Jopr agent is running.

It is possible to enable and disable metrics and also set the collection interval by accessing the **SCHEDULES** sub-menu, in **MONITOR** tab.

Appendix A. Java Development Kit (JDK): Installing, Configuring and Running

The **Mobicents Platform** is written in Java; therefore, before running any **Mobicents** server, you must have a working Java Runtime Environment (JRE) or Java Development Kit (JDK) installed on your system. In addition, the JRE or JDK you are using to run **Mobicents** must be version 5 or higher[1].

Should I Install the JRE or JDK? Although you can run **Mobicents** servers using the Java Runtime Environment, we assume that most users are developers interested in developing Java-based, **Mobicents**-driven solutions. Therefore, in this guide we take the tact of showing how to install the full Java Development Kit.

Should I Install the 32-Bit or the 64-Bit JDK, and Does It Matter? Briefly stated: if you are running on a 64-Bit Linux or Windows platform, you should consider installing and running the 64-bit JDK over the 32-bit one. Here are some heuristics for determining whether you would rather run the 64-bit Java Virtual Machine (JVM) over its 32-bit cousin for your application:

- Wider datapath: the pipe between RAM and CPU is doubled, which improves the performance of memory-bound applications when using a 64-bit JVM.
- 64-bit memory addressing gives virtually unlimited (1 exabyte) heap allocation. However large heaps affect garbage collection.
- Applications that run with more than 1.5 GB of RAM (including free space for garbage collection optimization) should utilize the 64-bit JVM.
- Applications that run on a 32-bit JVM and do not require more than minimal heap sizes will gain nothing from a 64-bit JVM. Barring memory issues, 64-bit hardware with the same relative clock speed and architecture is not likely to run Java applications faster than their 32-bit cousin.

Note that the following instructions detail how to download and install the 32-bit JDK, although the steps are nearly identical for installing the 64-bit version.

Downloading. You can download the Sun JDK 5.0 (Java 2 Development Kit) from Sun's website: *http://java.sun.com/javase/downloads/index_jdk5.jsp*. Click on the **Download** link next to "JDK 5.0 Update <*x*>" (where <*x*> is the latest minor version release number). On the next page, select your language and platform (both architecture—whether 32- or 64-bit—and operating

[1] At this point in time, it is possible to run most **Mobicents** servers, such as the JAIN SLEE, using a Java 6 JRE or JDK. Be aware, however, that presently the XML Document Management Server does not run on Java 6. We suggest checking the Mobicents web site, forums or discussion pages if you need to inquire about the status of running the XML Document Management Server with Java 6.

system), read and agree to the `Java Development Kit 5.0 License Agreement`, and proceed to the download page.

The Sun website will present two download alternatives to you: one is an RPM inside a self-extracting file (for example, `jdk-1_5_0_16-linux-i586-rpm.bin`), and the other is merely a self-extracting file (e.g. `jdk-1_5_0_16-linux-i586.bin`). If you are installing the JDK on Red Hat Enterprise Linux, Fedora, or another RPM-based Linux system, we suggest that you download the self-extracting file containing the RPM package, which will set up and use the SysV service scripts in addition to installing the JDK. We also suggest installing the self-extracting RPM file if you will be running **Mobicents** in a production environment.

Installing. The following procedures detail how to install the Java Development Kit on both Linux and Windows.

Procedure A.1. Installing the JDK on Linux

- Regardless of which file you downloaded, you can install it on Linux by simply making sure the file is executable and then running it:

```
~]$ chmod +x "jdk-1_5_0_<minor_version>-linux-<architecture>-rpm.bin"
~]$ ./"jdk-1_5_0_<minor_version>-linux-<architecture>-rpm.bin"
```

You Installed Using the Non-RPM Installer, but Want the SysV Service Scripts

If you download the non-RPM self-extracting file (and installed it), and you are running on an RPM-based system, you can still set up the SysV service scripts by downloading and installing one of the `-compat` packages from the JPackage project. Remember to download the `-compat` package which corresponds correctly to the minor release number of the JDK you installed. The compat packages are available from *ftp://jpackage.hmdc.harvard.edu/JPackage/1.7/generic/RPMS.non-free/*.

Important

You do not need to install a `-compat` package in addition to the JDK if you installed the self-extracting RPM file! The `-compat` package merely performs the same SysV service script set up that the RPM version of the JDK installer does.

Procedure A.2. Installing the JDK on Windows

- Using Explorer, simply double-click the downloaded self-extracting installer and follow the instructions to install the JDK.

Configuring. Configuring your system for the JDK consists in two tasks: setting the `JAVA_HOME` environment variable, and ensuring that the system is using the proper JDK (or JRE) using the `alternatives` command. Setting `JAVA_HOME` usually overrides the values for `java`, `javac` and `java_sdk_1.5.0` in `alternatives`, but we will set them all just to be safe and consistent.

Setting the `JAVA_HOME` Environment Variable on Generic Linux

After installing the JDK, you must ensure that the `JAVA_HOME` environment variable exists and points to the location of your JDK installation.

Setting the `JAVA_HOME` Environment Variable on Linux. You can determine whether `JAVA_HOME` is set on your system by `echo`ing it on the command line:

```
~]$ echo $JAVA_HOME
```

If `JAVA_HOME` is not set already, then you must set its value to the location of the JDK installation on your system. You can do this by adding two lines to your personal `~/.bashrc` configuration file. Open `~/.bashrc` (or create it if it doesn't exist) and add a line similar to the following one anywhere inside the file:

```
export JAVA_HOME="/usr/lib/jvm/jdk1.5.0_<version>"
```

You should also set this environment variable for any other users who will be running **Mobicents** (any environment variables `export`ed from `~/.bashrc` files are local to that user).

Setting `java`, `javac` and `java_sdk_1.5.0` Using the `alternatives` command

Selecting the Correct System JVM on Linux using `alternatives`. On systems with the `alternatives` command, including Red Hat Enterprise Linux and Fedora, you can easily choose which JDK (or JRE) installation you wish to use, as well as which `java` and `javac` executables should be run when called.

As the root user, call `/usr/sbin/alternatives` with the `--config java` option to select between JDKs and JREs installed on your system:

```
root@localhost ~]$ /usr/sbin/alternatives --config java

There are 3 programs which provide 'java'.

  Selection    Command
-----------------------------------------------
   1           /usr/lib/jvm/jre-1.5.0-gcj/bin/java
   2           /usr/lib/jvm/jre-1.6.0-sun/bin/java
*+ 3          /usr/lib/jvm/jre-1.5.0-sun/bin/java
```

```
Enter to keep the current selection[+], or type selection number:
```

In our case, we want to use the Sun JDK, version 5, that we downloaded and installed, to run the `java` executable. In the `alternatives` information printout above, a plus (+) next to a number indicates the one currently being used. As per `alternatives`' instructions, pressing **Enter** will simply keep the current JVM, or you can enter the number corresponding to the JVM you would prefer to use.

Repeat the procedure above for the `javac` command and the `java_sdk_1.5.0` environment variable, *as the root user*:

```
~]$ /usr/sbin/alternatives --config javac
```

```
~]$ /usr/sbin/alternatives --config java_sdk_1.5.0
```

Setting the `JAVA_HOME` Environment Variable on Windows

For information on how to set environment variables in Windows, refer to *http://support.microsoft.com/kb/931715*.

Testing. Finally, to make sure that you are using the correct JDK or Java version (5 or higher), and that the java executable is in your `PATH`, run the `java -version` command in the terminal from your home directory:

```
~]$ java -version
java version "1.5.0_16"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_16-b03)
Java HotSpot(TM) Client VM (build 1.5.0_16-b03, mixed mode, sharing)
```

Uninstalling. There is usually no reason (other than space concerns) to remove a particular JDK from your system, given that you can switch between JDKs and JREs easily using `alternatives`, and/or by setting `JAVA_HOME`.

Uninstalling the JDK on Linux. On RPM-based systems, you can uninstall the JDK using the `yum remove <jdk_rpm_name>` command.

Uninstalling the JDK on Windows. On Windows systems, check the JDK entry in the `Start` menu for an uninstall command, or use `Add/Remove Programs`.

Appendix B. Setting the JBOSS_HOME Environment Variable

The **Mobicents Platform** (**Mobicents**) is built on top of the **JBoss Application Server**. You do not need to set the `JBOSS_HOME` environment variable to run any of the **Mobicents Platform** servers *unless* `JBOSS_HOME` is *already* set.

The best way to know for sure whether `JBOSS_HOME` was set previously or not is to perform a simple check which may save you time and frustration.

Checking to See If JBOSS_HOME is Set on Unix. At the command line, `echo` **`$JBOSS_HOME`** to see if it is currently defined in your environment:

```
~]$ echo $JBOSS_HOME
```

The **Mobicents Platform** and most Mobicents servers are built on top of the **JBoss Application Server** (**JBoss Application Server**). When the **Mobicents Platform** or Mobicents servers are built *from source*, then `JBOSS_HOME` *must* be set, because the Mobicents files are installed into (or "over top of" if you prefer) a clean **JBoss Application Server** installation, and the build process assumes that the location pointed to by the `JBOSS_HOME` environment variable at the time of building is the **JBoss Application Server** installation into which you want it to install the Mobicents files.

This guide does not detail building the **Mobicents Platform** or any Mobicents servers from source. It is nevertheless useful to understand the role played by **JBoss AS** and `JBOSS_HOME` in the Mobicents ecosystem.

The immediately-following section considers whether you need to set `JBOSS_HOME` at all and, if so, when. The subsequent sections detail how to set `JBOSS_HOME` on Unix and Windows

Important

Even if you fall into the category below of *not needing* to set `JBOSS_HOME`, you may want to for various reasons anyway. Also, even if you are instructed that you do *not need* to set `JBOSS_HOME`, it is good practice nonetheless to check and make sure that `JBOSS_HOME` actually *isn't* set or defined on your system for some reason. This can save you both time and frustration.

You *DO NOT NEED* to set `JBOSS_HOME` if...

- ...you have installed the **Mobicents Platform** binary distribution.

- ...you have installed a Mobicents server binary distribution *which bundles **JBoss Application Server***.

You *MUST* set `JBOSS_HOME` if...

- ...you are installing the **Mobicents Platform** or any of the Mobicents servers *from source*.
- ...you are installing the **Mobicents Platform** binary distribution, or one of the Mobicents server binary distributions, which *do not* bundle **JBoss Application Server**.

Naturally, if you installed the **Mobicents Platform** or one of the Mobicents server binary releases which *do not* bundle **JBoss Application Server**, yet requires it to run, then you should install before setting `JBOSS_HOME` or proceeding with anything else.

Setting the JBOSS_HOME Environment Variable on Unix. The `JBOSS_HOME` environment variable must point to the directory which contains all of the files for the **Mobicents Platform** or individual Mobicents server that you installed. As another hint, this topmost directory contains a `bin` subdirectory.

Setting `JBOSS_HOME` in your personal `~/.bashrc` startup script carries the advantage of retaining effect over reboots. Each time you log in, the environment variable is sure to be set for you, as a user. On Unix, it is possible to set `JBOSS_HOME` as a system-wide environment variable, by defining it in `/etc/bashrc`, but this method is neither recommended nor detailed in these instructions.

Procedure B.1. To Set JBOSS_HOME on Unix...

1. Open the `~/.bashrc` startup script, which is a hidden file in your home directory, in a text editor, and insert the following line on its own line while substituting for the actual install location on your system:

   ```
   export JBOSS_HOME="/home/<username>/<path>/<to>/<install_directory>"
   ```

2. Save and close the `.bashrc` startup script.

3. You should `source` the `.bashrc` script to force your change to take effect, so that `JBOSS_HOME` becomes set for the current session[1].

   ```
   ~]$ source ~/.bashrc
   ```

4. Finally, ensure that `JBOSS_HOME` is set in the current session, and actually points to the correct location:

[1] Note that any other terminals which were opened prior to your having altered `.bashrc` will need to `source ~/.bashrc` as well should they require access to `JBOSS_HOME`.

Note

The command line usage below is based upon a binary installation of the **Mobicents Platform**. In this sample output, JBOSS_HOME has been set correctly to the *topmost_directory* of the **Mobicents** installation. Note that if you are installing one of the standalone **Mobicents** servers (with **JBoss AS** bundled!), then JBOSS_HOME would point to the *topmost_directory* of your server installation.

```
~]$ echo $JBOSS_HOME
/home/silas/
```

Setting the JBOSS_HOME Environment Variable on Windows. The JBOSS_HOME environment variable must point to the directory which contains all of the files for the Mobicents Platform or individual Mobicents server that you installed. As another hint, this topmost directory contains a bin subdirectory.

For information on how to set environment variables in recent versions of Windows, refer to *http://support.microsoft.com/kb/931715*.

Appendix C. Revision History

Revision History

Revision 1.0	Thu Dec 31 2009	AlexandreMendonça

Creation of the Mobicents Jopr JAIN SLEE Plugin User Guide.

Index